M 1

```
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRR      TTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRR      TTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRR      TTTTTTTTTTTTTT  LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT       LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT       LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT       LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT       LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT       LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT       LLL
BBBBBBBBBBBB     AAA         AAA     SSSSSSSSS     RRRRRRRRRRR            TTT       LLL
BBBBBBBBBBBB     AAA         AAA     SSSSSSSSS     RRRRRRRRRRR            TTT       LLL
BBBBBBBBBBBB     AAA         AAA     SSSSSSSSS     RRRRRRRRRRR            TTT       LLL
BBB         BBB  AAAAAAAAAAAAAAA              SSS  RRR   RRR              TTT       LLL
BBB         BBB  AAAAAAAAAAAAAAA              SSS  RRR   RRR              TTT       LLL
BBB         BBB  AAAAAAAAAAAAAAA              SSS  RRR   RRR              TTT       LLL
BBB         BBB  AAA         AAA              SSS  RRR      RRR           TTT       LLL
BBB         BBB  AAA         AAA              SSS  RRR      RRR           TTT       LLL
BBB         BBB  AAA         AAA              SSS  RRR      RRR           TTT       LLL
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT       LLLLLLLLLLLLLL
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT       LLLLLLLLLLLLLL
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT       LLLLLLLLLLLLLL
```

FILEID**BASUDFRL

L 11

```
BBBBBBBB      AAAAAA      SSSSSSSS  UU      UU  DDDDDDDD    FFFFFFFFFF  RRRRRRRR    LL
BBBBBBBB      AAAAAA      SSSSSSSS  UU      UU  DDDDDDDD    FFFFFFFFFF  RRRRRRRR    LL
BB      BB  AA      AA  SS          UU      UU  DD      DD  FF          RR      RR  LL
BB      BB  AA      AA  SS          UU      UU  DD      DD  FF          RR      RR  LL
BB      BB  AA      AA  SS          UU      UU  DD      DD  FF          RR      RR  LL
BB      BB  AA      AA  SS          UU      UU  DD      DD  FF          RR      RR  LL
BBBBBBBB    AA      AA    SSSSSS    UU      UU  DD      DD  FFFFFFFF    RRRRRRRR    LL
BBBBBBBB    AA      AA    SSSSSS    UU      UU  DD      DD  FFFFFFFF    RRRRRRRR    LL
BB      BB  AAAAAAAAAA          SS  UU      UU  DD      DD  FF          RR  RR      LL
BB      BB  AAAAAAAAAA          SS  UU      UU  DD      DD  FF          RR  RR      LL
BB      BB  AA      AA          SS  UU      UU  DD      DD  FF          RR    RR    LL
BB      BB  AA      AA          SS  UU      UU  DD      DD  FF          RR    RR    LL
BBBBBBBB    AA      AA  SSSSSSSS    UUUUUUUUUU  DDDDDDDD    FF          RR      RR  LLLLLLLLLL    ....
BBBBBBBB    AA      AA  SSSSSSSS    UUUUUUUUUU  DDDDDDDD    FF          RR      RR  LLLLLLLLLL    ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 MODULE BAS$$UDF_RL (                          ! BASIC list-directed input, UDF level
0                 IDENT = '1-075'               ! File: BASUDFRL.B32 Edit:MDL1075
0                 ) =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1
1 !++
1 ! FACILITY:      BASIC support library - not user callable
1 !
1 ! ABSTRACT:
1 !
1 !       This module implements BASIC read list-directed I/O statement
1 !       at the UDF level of abstraction.  This module calls the list-
1 !       directed record routines at the record level to read a record.
1 !
1 ! ENVIRONMENT:  User access mode, reentrant AST level or not
1 !
1 ! AUTHOR: Donald G. Petersen,   CREATION DATE: 23-MAR-78
1 !
1 ! MODIFIED BY:
1 !
1 !       DGP, 23-MAR-78 : VERSION 0
1 ! 1     - original
1 ! 1-02  - Change to JSB linkages.  DGP 14-Nov-78
1 ! 1-004 - Update copyright notice and add device names to REQUIRE
1 !           files.  JBS 29-NOV-78
1 ! 1-005 - Change REQUIRE file names from FOR... to OTS...  JBS 07-DEC-78
1 ! 1-006 - Change to new statement types for INPUT LINE and LINPUT. DGP
1 !           08-Dec-78
1 ! 1-007 - Change UDF_RL1 to use dispatch tables to get to REC level.  DGP
1 !           19-Dec-78
1 ! 1-008 - Add the necessary functionality to get INPUT LINE properly.  DGP
1 !           19-Dec-78
```

```
1 ! 1-009 - Bug fix.  DGP 20-Dec-78
1 ! 1-010 - Add support for longwords.  DGP 28-Dec-78
1 ! 1-011 - Add error signal to UDF_WL1 (BAS$K_ILLNUM).  DGP 28-Dec-78
1 ! 1-012 - Fix bug in Input integer (word).  DGP 02-Jan-79
1 ! 1-013 -  Change ISB$A_BUF_PTR, BUF_BEG, BUF_END to LUB. DGP 05-Jan-79
1 ! 1-014 - Make some "cleanup" edits based on the code review.
1 !          JBS for DGP.  09-JAN-1979
1 ! 1-015 - Correct some typos.  JBS 10-JAN-1979
1 ! 1-016 - Expand on some comments.  DGP 15-Jan-79
1 ! 1-017 - Add code to handle ^Z for INPUT LINE properly.  DGP 15-Jan-79
1 ! 1-018 - Fix bug in returning text string from GETFIELD. DGP 16-Jan-79
1 ! 1-019 - Change SIGNAL to STOP for ILLNUM in GETFIELD.  DGP 26-Jan-79
1 ! 1-020 - Use BASIOERR.REQ to define the I/O error codes.  JBS 20-FEB-1979
1 ! 1-021 - Modify GETFIELD to strip off leading and trailing spaces and tabs
1 !         from unquoted strings.  DGP 23-Feb-79
1 ! 1-022 - Change update of BUF_PTR for text in GETFIELD.  DGP 06-Mar-79
1 ! 1-023 - Strip all leading spaces and tabs from any text string before check-
1 !         ing for delimiting quotes.  DGP 15-Mar-79
1 ! 1-024 - Change PRINT_POS to longword.  DGP 19-Mar-79
1 ! 1-025 - Don't allow semicolon as numeric field separator on Input.  DGP
1 !         02-Apr-79
1 ! 1-026 - If this is not a terminal device, then ignore the prompt.  06-Apr-79
1 !         DGP
1 ! 1-027 - Change call to BAS$$STOP to BAS$$STOP_IO.  DGP 16-Apr-79
1 ! 1-028 - Change a few error messages.  DGP 07-May-79
1 ! 1-029 - Change OTS$S to STR$.  JBS 23-MAY-1979
1 ! 1-030 - BAS$$UDF_RL1 returns a status.  DGP 06-Jun-79
1 ! 1-031 - Fix up BAS$$UDF_RL1 to support MAT INPUT.  DGP 14-Jun-79
1 ! 1-032 - Use language-specific dispatch tables.  JBS 26-JUN-1979
1 ! 1-033 - Improve the comments.  DGP  28-Jun-79
1 ! 1-034 - Use ISB symbols for dispatch tables.  JBS 12-JUL-1979
1 ! 1-035 - Change calls to STR$COPY.  JBS 16-JUL-1979
1 ! 1-036 - Change from FOR$ input conversion routines to OTS$.  DGP 17-Jul-79
1 ! 1-037 - Remove reference to BAS$$SIGDIS_ERR.  JBS 01-AUG-1979
1 ! 1-038 - Set "don't round" flag for single precision floating when calling
1 !         the input conversion routine.  DGP 07-Aug-79
1 ! 1-039 - UDF_RL0 should dispatch to the REC level.  DGP 07-Aug-79
1 ! 1-040 - Set the prompt buffer size to 0 for MAT INPUT if REC level returns
1 !         a failure.  DGP 07-Aug-79
1 ! 1-041 - Strip off leading and trailing nulls from input.  DGP 29-Aug-79
1 ! 1-042 - Unconditionally clear the prompt buffer after every GET.  DGP 03-Sep-79
1 ! 1-043 - Switch the order of K_CRLF.  DGP 05-Sep-79
1 ! 1-044 - Increase K_WORK_STR_LEN to 512.  DGP 10-Sep-79
1 ! 1-045 - Fix bug in INPUT longwords with tabs and spaces.  DGP 10-Sep-79
1 ! 1-046 - Only look at low byte of RAB$L_STV for terminator.  DGP 18-Sep-79
1 ! 1-047 - Clear LUB$L_PRINT_POS just before the GET is done.  DGP 18-Sep-79
1 ! 1-048 - Prompting should be using LUB$B_PRINT_POS from LUB$A_BUDDY_PTR so
1 !         that CCPOS picks up the right value.  DGP 18-Sep-79
1 ! 1-049 - Check for comma after quoted string.  DGP 09-Oct-79
1 ! 1-050 - Include MAT LINPUT with those statement types which want to
1 !         read an entire line.  DGP 12-Oct-79
1 ! 1-051 - Another attempt at handling quoted strings properly.  DGP 18-Oct-79
1 ! 1-052 - Fix bug of input string that is only spaces, tabs, or nulls.
1 !         DGP 29-Oct-79
1 ! 1-053 - Pass the scale factor to the conversion routine.  DGP 25-Nov-79
1 ! 1-054 - Set V_EXP_LETTER for OTS$CVT_T_D.  DGP 04-DEC-79
1 ! 1-055 - Correct improper register declaration for scaling.  DGP 18-Dec-79
```

```
1 ! 1-056 - Call MTH$DINT_R3 instead of MTH$DFLOOR_R3 for scaling.  DGP 19-Dec-79
1 ! 1-057 - Signal DATA FORMAT ERROR instead of ILLEGAL NUMBER.  DGP  21-Jan-80
1 ! 1-058 - If this is READ or MAT READ then update the data pointer before
1 !         doing the conversion so that we are pointing at the next data
1 !         element.  DGP 22-Jan-80
1 ! 1-059 - Pick up escape sequences from RMS for INPUT LINE.  DGP 21-Feb-80
1 ! 1-060 - Do not set the cursor position unconditionally to zero.  DGP 04-Mar-80
1 ! 1-061 - If the terminator is an escape (altmode) and the terminating
1 !         sequence is of length 1, then transfer the escape character for
1 !         INPUT LINE.  RMS does not supply it at the end of the data anymore.
1 !         DGP 31-Mar-80
1 ! 1-062 - Fix the problem with inputing (READ,INPUT.....)
1 !         "abc"123,"xyz"  this should give an error because of 123.  FM 25-SEP-80
1 ! 1-063 - Enable INPUT and kind to take an input longer than K_STR_LEN bytes.
1 !         Terminal I/O is still restricted to 512 bytes.  FM 25-SEP-80
1 !         61A and 61B were put in the same packet.
1 ! 1-064 - Fix problem in above change.  A GTRU should be a GTR.  DGP 03-Feb-1981.
1 ! 1-065 - Change some occurences of CCB[LUB$L_PRINT_POS] TO TEMP_CCB[LUB$L_PRINT_POS].
1 !         Also, INPUT should cancel any outstanding PRINT formatcharacter
1 !         unless the INPUT was terminated by an escape. PLL 12-Jun-81
1 ! 1-066 - A case statement in GETFIELD modified to always return a value
1 !         so that the BLISS compiler does not give an error message.
1 !         PLL 1-Jul-81
1 ! 1-067 - 64k bytes of data causes a premature "out of data" message because
1 !         SCANC length is limited to 16 bits.  Make sure the length always looks
1 !         = or < 64k to GETFIELD.  PLL 23-Jul-81
1 ! 1-068 - Add support for byte, g floating, and h floating.  PLL 24-Aug-81
1 ! 1-069 - Add support for packed decimal.  PLL 5-Oct-81
1 ! 1-070 - More edits for packed decimal.  PLL 29-Dec-81
1 ! 1-071 - Correct a typo in range check on byte.  PLL 9-Mar-1982
1 ! 1-072 - Before calling BAS$CVT_T_P, check the decimal rounding/truncation
1 !         bit in the Basic frame.
1 ! 1-073 - Add support for ANSI INPUT.  Although input is always from a
1 !         terminal, errors should cause the entire statement to be re-
1 !         started not just the specific element.  This means that $GET
1 !         occurs at the 0 level rather than level 1.  PLL 29-Jul-1982
1 ! 1-074 - ANSI INPUT of a single element should signal 'too little data',
1 !         not supply the default for the data type.  PLL 27-Sep-1982
1 ! 1-075 - allow for terminator space when allocating space for WORK_STR.
1 !         MDL 25-Apr-1984
1 !--
```

```
1
1 !
1 ! SWITCHES:
1 !
1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
1 !
1 !       LINKAGES:
1 !
1
1 REQUIRE 'RTLIN:OTSLNK';            ! define all linkages
1
1 !
1 ! TABLE OF CONTENTS:
1 !
1
1 FORWARD ROUTINE
1
1     !
1     ! UDF routines
1     !
1     BAS$$UDF_RL0: JSB_UDF0 NOVALUE,
1     BAS$$UDF_RL1: CALL_CCB,
1     UDF_RL1_HANDLER,
1     BAS$$UDF_RL9: JSB_UDF9 NOVALUE,
1
1     !
1     ! routine used by  BAS$$UDF_RL1
1     !
1     GETFIELD: CALL_CCB;
1 !
1 ! INCLUDE FILES:
1 !
1 REQUIRE 'RTLML:BASPAR';             ! BASIC intermodule parameters
1 REQUIRE 'RTLIN:BASFRAME';           ! BASIC frame offsets
1 REQUIRE 'RTLML:OTSISB';             ! I/O statement block
1 REQUIRE 'RTLML:OTSLUB';             ! Logical Unit Block
1 REQUIRE 'RTLIN:OTSMAC';             ! Macros
1 REQUIRE 'RTLIN:RTLPSECT';           ! Define DECLARE_PSECTS macro
1 REQUIRE 'RTLIN:BASIOERR';           ! Define I/O error codes.
1 LIBRARY 'RTLSTARLE';                ! STARLET library for macros and symbols
1
1 !
1 ! MACROS:
1 !
1 !       NONE
1 !
1 !
1 ! EQUATED SYMBOLS:
1 !
1
1 LITERAL
1
1     K_WORK_STR_LEN = 512,                   ! length of work area for parsing input.
1     K_NULL       = 0,                       ! types of constants which may appear in input record
1     K_CR         = %X'0D',                  ! ASCII <cr>
1     K_ESC        = %X'1B',                  ! ASCII <esc>
1     K_SP         = %X'20',                  ! ASCII <sp>
```

```
    K_TAB       = 9;                          ! ASCII TAB

BUILTIN
    CVTSP,
    SCANC;

!
! PSECT declarations
!
    DECLARE_PSECTS (BAS);                      ! declare PSECTS for BAS$ facility

!
! OWN STORAGE:
!       NONE

!
! EXTERNAL REFERENCES:
!

EXTERNAL LITERAL
    BAS$K_DATFORERR:UNSIGNED (8),          ! Data format error
    BAS$K_ILLNUM:UNSIGNED (8),             ! Illegal number
    BAS$K_ENDFILDEV:UNSIGNED (8),          ! End of file on device
    BAS$K_MAXMEMEXC:UNSIGNED (8),          ! Maximum memory exceeded
    BAS$K_PROLOSSOR:UNSIGNED (8),          ! Program lost sorry
    BAS$K_TOOLITDAT:UNSIGNED (8);          ! Too little data (ANSI only)

EXTERNAL
    BAS$$AA_REC_PR0 : VECTOR,              ! Dispatch table for REC init.
    BAS$$AA_REC_PR1 : VECTOR,              ! Dispatch table for REC level
    OTS$$A_CUR_LUB: ADDRESSING_MODE (GENERAL), ! address of currently active LUB/ISB/RAB
    BAS$HANDLER;                           ! just need the address of this

EXTERNAL ROUTINE
    MTH$DINT,                     ! Remove fraction after scaling
    BAS$$STOP_IO,                 ! signal fatal errors
    BAS$$SIGNAL_IO,               ! signal an error
    LIB$CVTDF,                    ! convert double to floating
    STR$COPY_DX,                  ! Copy a string by descriptor
    BAS$OUT_T_DX_S:  NOVALUE,     ! output a text string
    BAS$CVT_T_P,                  ! convert text to packed decimal
    !
    ! conversion routines
    !
    OTS$CVT_TI_L,                 ! convert ASCII to internal 32 bit integer
    OTS$CVT_T_D,                  ! convert ASCII to internal double precision
    OTS$CVT_T_G,                  ! convert ASCII to internal g floating
    OTS$CVT_T_H,                  ! convert ASCII to internal h floating
    !
    ! record level routines for list-directed input
    !
    BAS$$REC_RSL0: JSB_REC0 NOVALUE,    ! initialize Input record level
    BAS$$REC_RSL9: JSB_REC9 NOVALUE,    ! end of Input record level
    LIB$GET_VM,                         ! get virtual memory
    LIB$FREE_VM,                        ! free virtual memory
    LIB$MATCH_COND;                     ! match the condition value
```

```
1 GLOBAL ROUTINE BAS$$UDF_RL0 (
1         FORMAT_ADR
1         ): JSB_UDF0 NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Perform UDF level read list-directed I/O initialization.
1 !       Initialize module "own" storage in the ISB.
1 !       Call record level processor to get first input record.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       FORMAT_ADR.rl.r         Not used
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       OTS$$A_CUR_LUB          Pointer to current logical unit block (LUB)
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2     EXTERNAL REGISTER
2         CCB: REF BLOCK[, BYTE];
2
2     !+
2     ! Call record level routine to read the first record.
2     ! The buffer pointers are initialized based on whether the device is
2     ! a terminal or not
2     !-
2
2     !+
2     ! If this is an ANSI INPUT, the RECO level will ask for input.  So
2     ! put out the standard prompt.  Note:  ANSI has no files, so INPUT
2     ! will always be from a terminal.
2     !-
2
2         IF .CCB [LUB$V_ANSI]
2         THEN
3             BEGIN
3             LOCAL
3                 TDSC: VECTOR [2];
3             BIND
3                 D_PROMPT = UPLIT ('? ');
```

```
;  329        1623  3              TDSC[0] = %CHARCOUNT ('? ');
;  330        1624  3              TDSC[1] = D_PROMPT;
;  331        1625  3              BAS$OUT_T_DX_S(TDSC);
;  332        1626  2              END;
;  333        1627  2
;  334        1628  2       JSB_RECO (BAS$$AA_REC_PRO + .BAS$$AA_REC_PRO [.CCB [ISB$B_STTM_TYPE] - ISB$K_BASSTTYLO + 1]);
;  335        1629  2
;  336        1630  1       END;


                                                              .TITLE   BAS$$UDF_RL
                                                              .IDENT   \1-075\

                                                              .PSECT   _BAS$CODE,NOWRT,  SHR,  PIC,2

                              00  00  20  3F  00000 P.AAA:    .ASCII   \? \<0><0>                          ;

                                                    D_PROMPT=           P.AAA
                                                              .EXTRN   BAS$K_DATFORERR
                                                              .EXTRN   BAS$K_ILLNUM, BAS$K_ENDFILDEV
                                                              .EXTRN   BAS$K_MAXMEMEXC
                                                              .EXTRN   BAS$K_PROLOSSOR
                                                              .EXTRN   BAS$K_TOOLITDAT
                                                              .EXTRN   BAS$$AA_REC_PRO
                                                              .EXTRN   BAS$$AA_REC_PR1
                                                              .EXTRN   OTS$$A_CUR_LUB, BAS$HANDLER
                                                              .EXTRN   MTH$DINT, BAS$$STOP_IO
                                                              .EXTRN   BAS$$SIGNAL_IO, LIB$CVTDF
                                                              .EXTRN   STR$COPY_DX, BAS$OUT_T_DX_S
                                                              .EXTRN   BAS$CVT_T_P, OTS$CVT_TI_L
                                                              .EXTRN   OTS$CVT_T_D, OTS$CVT_T_G
                                                              .EXTRN   OTS$CVT_T_H, BAS$$REC_RSL0
                                                              .EXTRN   BAS$$REC_RSL9, LIB$GET_VM
                                                              .EXTRN   LIB$FREE_VM, LIB$MATCH_COND

                         5E              08  C2 00000 BAS$$UDF_RL0::
                                                              SUBL2    #8, SP                              ; 1566
             11      A1  AB              04  E1 00003         BBC      #4, -95(CCB), 1$                    ; 1616
                         6E              02  D0 00008         MOVL     #2, TDSC                            ; 1623
             04      AE        EE        AF  9E 0000B         MOVAB    D_PROMPT, TDSC+4                    ; 1624
                                         5E  DD 00010         PUSHL    SP                                  ; 1625
          00000000G  00                  01  FB 00012         CALLS    #1, BAS$OUT_T_DX_S
                         50        FF71  CB  9A 00019 1$:     MOVZBL   -143(CCB), R0                       ; 1628
                         50 00000000G0040    D0 0001E         MOVL     BAS$$AA_REC_PRO-104[R0], R0         ;
                            00000000G0040    16 00026         JSB      BAS$$AA_REC_PRO[R0]                 ;
                         5E              08  C0 0002D         ADDL2    #8, SP                              ; 1630
                                             05 00030         RSB                                          ;

; Routine Size:  49 bytes,    Routine Base:  _BAS$CODE + 0004
```

```
1 GLOBAL ROUTINE BAS$$UDF_RL1 (
1               ELEM_TYPE,
1               ELEM_SIZE,
1               ELEM_ADR,
1               FORMAT
1               )
1               : CALL_CCB =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !             Return the next input value to the user I/O list element.
1 !             The value obtained from the input record is converted to
1 !             the type of the list element.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !             ELEM_TYPE.rlu.v         Type code of user I/O list element
1 !             ELEM_SIZE.rlu.v         Size of the list element
1 !             ELEM_ADR.rlu.r          Adr of where to store the element
1 !                                     Points to a descriptor for a string
1 !             FORMAT.rlu.v            Format character following a Prompt string
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !             OTS$$A_CUR_LUB          Pointer to current logical unit block (LUB)
1 !             LUB$L_PRINT_POS         Internal cursor position
1 !             LUB$V_UNIT_0            flag to indicate terminal on unit 0
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !             LUB$L_PRINT_POS         internal cursor position
1 !             RAB$B_PSZ               size of the Prompt buffer
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !             NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !             SIGNALS various errors for input incompatibility and not enough
1 !             input data.
1 !             If this is not a terminal device, then ignore any prompts.
1 !
1 !             NOTICE : All terminal device files are allocated the static buffer for
1 !                      parsing, i.e. no VM is allocated for them (because at the
1 !                      time this routine is called we don't know how large of input
1 !                      we have!!). This means that the maximum terminal device input
1 !                      is K_WORK_STR_LEN, anything over this will write over the
1 !                      stack.
1 !--
1
1       !+
1       ! Be aware that there are two exit points in this routine.  One is from
1       ! the Prompt handling section and the other is from the Input handling section
1       !-
```

```
1
2      BEGIN
2      MAP
2          ELEM_ADR: REF VECTOR;
2      LOCAL
2          BYTES_NEEDED: INITIAL(0),              ! workspace needed
2          WORKSPACE: VECTOR [ K_WORK_STR_LEN , BYTE ],! if input is K_WORK_STR_LEN or less
2                                                 ! use this space through CHARCONS.
2          CHARCONS: REF VECTOR [ , BYTE ],! The space where the parsing of input takes place.
2          D_VALUE: VECTOR[4],                    ! holds binary equivalent of input char.
2                                                 ! for numerics
2          TEMP_CCB : REF BLOCK [, BYTE],         ! temporary CCB
2          DSC: BLOCK [8,BYTE],                   ! Descriptor of parsed element for strings
2                                                 ! need a local descriptor because if a
2                                                 ! static desc. was passed, the values in
2                                                 ! it are not to be changed.
2          UNWIND_VM_SIZE : VOLATILE,             ! Size of VM allocated.
2          UNWIND_VM_ADDR : VOLATILE,             ! Address of VM allocated for input buffer
2                                                 ! This buffer is allocated if input size
2                                                 ! is greater than 512.
2          UNWIND_CCB     : VOLATILE;             ! CCB for the handler.
2
2      LITERAL
2          K_ESC = %X'1B';           ! ASCII for escape
2
2      EXTERNAL REGISTER
2          CCB: REF BLOCK[,BYTE];
2
2      BUILTIN
2          FP;
2
2  !+
2  ! Set up a handler for this routine so in case of unwind we can deallocate VM,
2  ! if any was allocated.
2  !-
2      ENABLE UDF_RL1_HANDLER ( UNWIND_VM_SIZE , UNWIND_VM_ADDR , UNWIND_CCB );
2
2  !+
2  ! determine how much workspace is needed.  this is the number of bytes in
2  ! the buffer plus the number of bytes in the terminator.
2  !-
3      BYTES_NEEDED = ( (.CCB [LUB$A_BUF_END] - .CCB [LUB$A_BUF_PTR]) +
4                      (SELECTONEU .CCB [RAB$W_STV0] OF
4                       SET
4                          [K_ESC]:          .CCB [RAB$W_STV2];
4                          [K_CR]:           2;
4                          [OTHERWISE]:      0;
2                       TES) );
2  !+
2  ! If space needed for parsing is greater than K_WORK_STR_LEN then we use VM, otherwise
2  ! we use the static storage allocated in WORKSPACE.
2  !-
2      IF .BYTES_NEEDED GTR K_WORK_STR_LEN
2      THEN
3          BEGIN
3          UNWIND_VM_SIZE = .BYTES_NEEDED;
3          UNWIND_CCB = .CCB;
```

```
3           IF NOT LIB$GET_VM ( UNWIND_VM_SIZE , UNWIND_VM_ADDR ) THEN BAS$$STOP_IO (BAS$K_MAXMEMEXC);
3           CHARCONS = .UNWIND_VM_ADDR;
3           END
2       ELSE
2           CHARCONS = WORKSPACE;
2   !+
2   ! Load up TEMP_CCB with a pointer to the complementary data base for PRINT.
2   !-
2       TEMP_CCB = .CCB [LUB$A_BUDDY_PTR];
2
2       IF .FORMAT GTR 0
2       THEN
3           BEGIN
3
3           !+
3           ! Check to see if this is a terminal device.  If it is, then process the
3           ! prompt; otherwise, just return.
3           !-
3
3           IF .CCB [LUB$V_TERM_DEV]
3           THEN
4               BEGIN
4
4           !+
4           ! Prompt
4           !-
4
4           LOCAL
4               RDSC: BLOCK [8, BYTE];          ! Resultant descriptor from Prompt processing
4           LITERAL
4               K_PRINT_ZONE_SZ = 14,           ! Print zone size
4               K_CRLF = %X'0A0D';              ! ASCII codes for carriage return-line feed
4
4           RDSC[DSC$A_POINTER] = .CCB[RAB$L_PBF] + .CCB[RAB$B_PSZ];
4
4           !+
4           ! adjust the internal cursor position and the resultant string
4           ! length as determined by the data type and the format character
4           !-
4
4           CASE .FORMAT
4           FROM BAS$K_SEMI_FORM TO BAS$K_NO_FORM OF
4           SET
4           [BAS$K_SEMI_FORM]:
5               BEGIN
5               CCB[ISB$V_P_FORM_CH] = BAS$K_SEMI_FORM;
5               RDSC[DSC$W_LENGTH] = .ELEM_SIZE;
5               TEMP_CCB [LUB$L_PRINT_POS] = .ELEM_SIZE + .TEMP_CCB [LUB$L_PRINT_POS];
4               END;
4           [BAS$K_COMMA_FOR]:
5               BEGIN
5               CCB[ISB$V_P_FORM_CH] = BAS$K_COMMA_FOR;
8               RDSC[DSC$W_LENGTH] = .ELEM_SIZE + (K_PRINT_ZONE_SZ - ((.TEMP_CCB [LUB$L_PRINT_POS] + .ELEM_SIZE)
5                   MOD K_PRINT_ZONE_SZ));
5               TEMP_CCB [LUB$L_PRINT_POS] = .TEMP_CCB[LUB$L_PRINT_POS] + .RDSC[DSC$W_LENGTH];
4               END;
4           [BAS$K_NO_FORM]:
```

```
5                   BEGIN
5
5                   !+
5                   ! Need to leave room for carriage control
5                   !-
5
5                   RDSC[DSC$W_LENGTH] = .ELEM_SIZE + 2;
5                   CCB[ISB$V_P_FORM_CH] = BAS$K_NO_FORM;
5                   TEMP_CCB[[UB$L_PRINT_POS] = 0;
4                   END;
4               TES;
4
4               !+
4               ! Set the address for the destination of the Prompt.  Update the RAB
4               ! Prompt Buffer Size
4               !-
4
4               CCB[RAB$B_PSZ] = .CCB[RAB$B_PSZ] + .RDSC[DSC$W_LENGTH];
4               RDSC[DSC$B_CLASS] = DSC$K_CLASS_S;
4               CH$COPY (.ELEM_SIZE, .(.ELEM_ADR+4), ' ', .RDSC[DSC$W_LENGTH], .RDSC[DSC$A_POINTER]);
4               IF .FORMAT EQLU BAS$K_NO_FORM
4               THEN
4                   (.RDSC[DSC$A_POINTER] + .ELEM_SIZE)<0, 16> = K_CRLF;
3               END;
3               RETURN 1;
2               END;
2
2           !+
2           ! This section is concerned with inputting a value
2           ! GETFIELD will attempt to parse another field out of the INPUT stream based
2           ! on the data type.  If a data field cannot be found (empty buffer)
2           ! then a failure
2           ! status is returned.  If a data field is found then a
2           !conversion, for numerics,
2           ! is done and if a conversion error occurs, the error number is put into the
2           ! LUB.  For a string, the descriptor passed to GETFIELD is updated to point to
2           ! the parsed string and the length field is updated.
2           !-
2
3           IF NOT (GETFIELD(
3
3                       !+
3                       ! Pass the a reference to a quadword for a numeric quantity and
3                       ! a pointer to a descriptor for a string
3                       !-
3
4                       (CASE .ELEM_TYPE
4                       FROM DSC$K_DTYPE_B TO DSC$K_DTYPE_H OF
4                       SET
4                       [DSC$K_DTYPE_B, DSC$K_DTYPE_W, DSC$K_DTYPE_L, DSC$K_DTYPE_F,
4                        DSC$K_DTYPE_D, DSC$K_DTYPE_G, DSC$K_DTYPE_H]:
4                           D_VALUE;
4                       [DSC$K_DTYPE_T, DSC$K_DTYPE_P] :
4                           DSC;
4                       [INRANGE, OUTRANGE]:
4                           !+
4                           ! Data types which are not yet supported
;  4                      !-
;  4                      0
;  4                  TES
;  3                  ),
;  3          .ELEM_TYPE, .CHARCONS))
;  2      THEN
;  3          BEGIN
;  3          !+
;  3          ! Try to get another record.  Device type checking (forcible or nonforcible) is performed at
;  3          ! the REC level before a GET is attempted.
;  3          !-
;  3          IF .CCB[LUB$V_UNIT_0] AND NOT .CCB [LUB$V_ANSI]
;  3          THEN
;  3              !+
;  3              ! Insert the BASIC default prompt if on unit 0
;  3              !-
;  3
;  4              BEGIN
;  4              LOCAL
;  4                  TDSC: VECTOR [2];
;  4              BIND
;  4                  D_PROMPT = UPLIT ('? ');
;  4              TDSC[0] = %CHARCOUNT ('? ');
;  4              TDSC[1] = D_PROMPT;
;  4              BAS$OUT_T_DX_S(TDSC);
;  3              END;
;  3
;  3          !+
;  3          ! Dispatch to the appropriate REC level routine.  If INPUT then get
;  3          ! another record.  If READ then signal an error - should not be out
;  3          ! of data. If this is a MAT INPUT, try to get another record and pass the status
;  3          ! back to the UPI level.  Status is determined by whether the current
;  3          ! record ends with a continuation character.  Clear LUB$L_PRINT_POS thru
;  3          ! BUDDY_PTR so that this INPUT will not affect later PRINTs or prompting
;  3          ! if there is an error on this GET.
;  3          ! NOTE:  There is a RETURN here in the middle of the routine.
;  3          !-
;  3
;  4          IF (NOT (JSB_REC1 (BAS$$AA_REC_PR1 + .BAS$$AA_REC_PR1[.CCB[ISB$B_STTM_TYPE] - ISB$K_BASSTTYLO + 1]))
;  3          THEN
;  3  !+
;  3  ! Clear the Prompt buffer which has been loaded in case another GET was going to
;  3  ! be done.  If it is not cleared, then I/O END will print it out (10 INPUT 'foo'
;  3  ! ).  MAT INPUT is different, because the RTL asks for more data if it is avail-
;  3  ! able.  The other types of Input demand more data.  Therefore, the GET for MAT
;  3  ! INPUT is only done if the continuation flag is set signifying that the last
;  3  ! record ended in an '&'.
;  3  !-
;  4              BEGIN
;  4              CCB [RAB$B_PSZ] = 0;
;  4              RETURN 0;
;  3              END;
;  3  !+
;  3  ! Unconditionally clear the prompt buffer so that a RESUME with no line number
;  3  ! which restarts an INPUT statement will not keep concatenating prompt strings.
;  3  !-
;  3          CCB [RAB$B_PSZ] = 0;
```

```
      !+
      ! Now that another record has been gotten, call GETFIELD again and ignore
      ! the return status because it is assumed that failure to return something
      ! is impossible.
      !-

      GETFIELD(
              (CASE .ELEM_TYPE
              FROM DSC$K_DTYPE_B TO DSC$K_DTYPE_H OF
              SET
              [DSC$K_DTYPE_B, DSC$K_DTYPE_W, DSC$K_DTYPE_L, DSC$K_DTYPE_F,
               DSC$K_DTYPE_D, DSC$K_DTYPE_G, DSC$K_DTYPE_H]:
                  D_VALUE;
              [DSC$K_DTYPE_T, DSC$K_DTYPE_P]:
                  DSC;
              [INRANGE, OUTRANGE]:
                  !+
                  ! Data types which are not yet supported
                  !-
                  0
              TES
              ),
          .ELEM_TYPE, .CHARCONS)
      END;

  !+
  ! Store the converted Input data into its new home based on the data type
  !-

  CASE .ELEM_TYPE
  FROM DSC$K_DTYPE_B TO DSC$K_DTYPE_H OF
  SET
  [INRANGE, OUTRANGE]:
  !+
  ! Data types which are not supported
  !-
  0;
  [DSC$K_DTYPE_B]:
      !+
      ! Byte
      !-
      BEGIN
      MAP
          ELEM_ADR: REF VECTOR[, BYTE];
      ELEM_ADR[0] = .D_VALUE;
      END;
  [DSC$K_DTYPE_W]:
      !+
      ! Integer
      !-
      BEGIN
      MAP
          ELEM_ADR: REF VECTOR[, WORD];
      ELEM_ADR[0] = .D_VALUE;
      END;
```

```
	[DSC$K_DTYPE_L, DSC$K_DTYPE_F]:
	    !+
	    ! Longword integer or single precision floating point
	    !-
	    ELEM_ADR[0] = .D_VALUE;
	[DSC$K_DTYPE_D, DSC$K_DTYPE_G]:
	    !+
	    ! Double precision floating point or g floating
	    !-
	    BEGIN
	    ELEM_ADR[0] = .D_VALUE[0];
	    ELEM_ADR[1] = .D_VALUE[1];
	    END;
	[DSC$K_DTYPE_H]:
	    !+
	    ! H floating
	    !-
	    BEGIN
	    ELEM_ADR[0] = .D_VALUE[0];
	    ELEM_ADR[1] = .D_VALUE[1];
	    ELEM_ADR[2] = .D_VALUE[2];
	    ELEM_ADR[3] = .D_VALUE[3];
	    END;
	[DSC$K_DTYPE_T]:
	    !+
	    ! Character string - ELEM_ADR contains the address of the descriptor
	    !-
	    BEGIN
	    DSC[DSC$A_POINTER] = .CHARCONS;
	    DSC[DSC$B_CLASS] = DSC$K_CLASS_S;
	    DSC[DSC$B_DTYPE] = DSC$K_DTYPE_T;
	    ! ***** Change to LIB$SCOPY to inhibit signalling ******
	    STR$COPY_DX (.ELEM_ADR, DSC);
	    IF .(.DSC[DSC$A_POINTER])<0,8> EQLU BAS$K_CONTROL_Z
	    THEN
		BEGIN

		!+
		! This ^Z has been deferred until now so that it could get stored into
		! the users buffer for error handling as required by Basic.  Now is
		! the proper time to signal the error.
		!-

		CCB[RAB$B_PSZ] = 0;
		BAS$$STOP_IO(BAS$K_ENDFILDEV);
		END;
	    END;
	[DSC$K_DTYPE_P]:
	    !+
	    ! Packed decimal string - ELEM_ADR contains the address of the descriptor
	    !-
	    BEGIN
	    LOCAL
		STATUS,
		FLAGS,
		FMP : REF BLOCK [0,BYTE] FIELD (BSF$FCD);
```

```
;  737          2030  3             LITERAL
;  738          2031  3                 V_DONT_ROUND = 1^3;
;  739          2032  3
;  740          2033  3             DSC[DSC$A_POINTER] = .CHARCONS;
;  741          2034  3             DSC[DSC$B_CLASS] = DSC$K_CLASS_S;
;  742          2035  3             DSC[DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  743          2036  3             !+
;  744          2037  3             ! Call a conversion routine which will handle the semantics of converting
;  745          2038  3             ! text to packed decimal.  Pass the decimal round/truncate flag from the
;  746          2039  3             ! Basic frame as the flags parameter.
;  747          2040  3             !-
;  748          2041  3             FMP = .FP;
;  749          2042  3
;  750          2043  3             DO
;  751          2044  4                 BEGIN                       ! search for a Basic frame
;  752          2045  4                 FMP = .FMP [BSF$A_SAVED_FP];
;  753          2046  4                 END
;  754          2047  4             UNTIL (.FMP [BSF$A_HANDLER] EQLA BAS$HANDLER OR
;  755          2048  3                     .FMP EQL 0);
;  756          2049  3
;  757          2050  3             IF (.FMP NEQ 0) AND (.FMP [BSF$W_FCD_FLAGS] AND BSF$M_FCD_RND) NEQ 0
;  758          2051  3             THEN
;  759          2052  3                 FLAGS = 0
;  760          2053  3             ELSE
;  761          2054  3                 FLAGS = V_DONT_ROUND;       ! set flags according to frame bit
;  762          2055  3
;  763          2056  3             STATUS = BAS$CVT_T_P (DSC, (.ELEM_ADR), .FLAGS);
;  764          2057  3             IF NOT .STATUS THEN BAS$$STOP_IO (BAS$K_DATFORERR);
;  765          2058  3             END
;  766          2059  2         TES;
;  767          2060  2         CCB[RAB$B_PSZ] = 0;
;  768          2061  2         IF (.CCB[RAB$W_STV0] NEQ K_ESC) THEN TEMP_CCB[LUB$V_FORM_CHAR] = 0;
;  769          2062  2     !+
;  770          2063  2     ! If we have allocated VM for the parsing space then deallocate it here.
;  771          2064  2     !-
;  772          2065  3         IF ( .CHARCONS NEQA WORKSPACE )
;  773          2066  2         THEN
;  774          2067  3             BEGIN
;  775          2068  3             IF NOT LIB$FREE_VM ( UNWIND_VM_SIZE , UNWIND_VM_ADDR )
;  776          2069  3             THEN
;  777          2070  4                 BEGIN
;  778          2071  4                 UNWIND_VM_SIZE = 0;
;  779          2072  4                 BAS$$STOP_IO (BAS$K_PROLOSSOR);
;  780          2073  3                 END;
;  781          2074  2             END;
;  782          2075  2         RETURN 1;
;  783          2076  1         END;



                                          00035          .BLKB    3
                      00  00  20  3F      00038  P.AAB:  .ASCII   \? \<0><0>                            ;

                                                 D_PROMPT=             P.AAB


                                    07FC  00000          .ENTRY   BAS$$UDF_RL1, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 1631
```

```
                                                                        R10                                          ;
                          5A 00000000G   00   9E 00002          MOVAB   BAS$$STOP_IO, R10                            ;
                          5E      FDD4   CE   9E 00009          MOVAB   -556(SP), SP                                 ;
                                         51   D4 0000E          CLRL    BYTES_NEEDED                                 ; 1689
                                   08    AE   7C 00010          CLRQ    UNWIND_CCB                                   ;
                                   10    AE   D4 00013          CLRL    UNWIND_VM_SIZE                               ;
                          6D      02CD   CF   DE 00016          MOVAL   49$, (FP)                                    ;
              52      B4  AB        B0   AB   C3 0001B          SUBL3   -80(CCB), -76(CCB), R2                       ; 1729
                          50        0C   AB   3C 00021          MOVZWL  12(CCB), R0                                  ; 1730
                          1B             50   B1 00025          CMPW    R0, #27                                      ; 1732
                          06             12   00028             BNEQ    1$                                           ;
                          50        0E   AB   3C 0002A          MOVZWL  14(CCB), R0                                  ;
                          0C             11   0002E             BRB     3$                                           ;
                          0D             50   B1 00030 1$:      CMPW    R0, #13                                      ; 1733
                          05             12   00033             BNEQ    2$                                           ;
                          50             02   D0 00035          MOVL    #2, R0                                       ;
                          02             11   00038             BRB     3$                                           ;
                                         50   D4 0003A 2$:      CLRL    R0                                           ; 1734
              51          52             50   C1 0003C 3$:      ADDL3   R0, R2, BYTES_NEEDED                         ; 1730
                  00000200  8F           51   D1 00040          CMPL    BYTES_NEEDED, #512                           ; 1740
                          25             15   00047             BLEQ    5$                                           ;
                     10   AE             51   D0 00049          MOVL    BYTES_NEEDED, UNWIND_VM_SIZE                 ; 1743
                     08   AE             5B   D0 0004D          MOVL    CCB, UNWIND_CCB                              ; 1744
                                   0C    AE   9F 00051          PUSHAB  UNWIND_VM_ADDR                               ; 1745
                                   14    AE   9F 00054          PUSHAB  UNWIND_VM_SIZE                               ;
                  00000000G  00          02   FB 00057          CALLS   #2, LIB$GET_VM                               ;
                          07             50   E8 0005E          BLBS    R0, 4$                                       ;
                          7E       00G   8F   9A 00061          MOVZBL  #BAS$K_MAXMEMEXC, -(SP)                      ;
                          6A             01   FB 00065          CALLS   #1, BAS$$STOP_IO                             ;
                          59        0C   AE   D0 00068 4$:      MOVL    UNWIND_VM_ADDR, CHARCONS                     ; 1746
                          04             11   0006C             BRB     6$                                           ; 1740
                          59        2C   AE   9E 0006E 5$:      MOVAB   WORKSPACE, CHARCONS                          ; 1749
                          58        B8   AB   D0 00072 6$:      MOVL    -72(CCB), TEMP_CCB                           ; 1753
                          57        10   AC   D0 00076          MOVL    FORMAT, R7                                   ; 1755
                                         03   14 0007A          BGTR    7$                                           ;
                                       0083   31 0007C          BRW     14$                                          ;
              7B      FE  AB             05   E1 0007F 7$:      BBC     #5, -2(CCB), 13$                             ; 1764
                          50        34   AB   9A 00084          MOVZBL  52(CCB), R0                                  ; 1778
                     04   AE        30 BB40   9E 00088          MOVAB   @48(CCB)[R0], RDSC+4                         ;
                          56        08   AC   D0 0008E          MOVL    ELEM_SIZE, R6                                ; 1791
                          50        C8   A8   9E 00092          MOVAB   -56(TEMP_CCB), R0                            ; 1792
              02          01             57   CF 00096          CASEL   R7, #1, #2                                   ; 1785
            0038        0014           0006      0009A 8$:      .WORD   9$-8$,-                                      ;
                                                                        10$-8$,-                                     ;
                                                                        11$-8$                                       ;
      96   AB   02          00           01   F0 000A0 9$:      INSV    #1, #0, #2, -106(CCB)                        ; 1790
                            6E           56   B0 000A6          MOVW    R6, RDSC                                     ; 1791
                            60           56   C0 000A9          ADDL2   R6, (R0)                                     ; 1792
                                         2E   11 000AC          BRB     12$                                          ; 1785
      96   AB   02          00           02   F0 000AE 10$:     INSV    #2, #0, #2, -106(CCB)                        ; 1796
                51          60           56   C1 000B4          ADDL3   R6, (R0), R1                                 ; 1797
           7E   00          51           01   7A 000B8          EMUL    #1, R1, #0, -(SP)                            ; 1798
           51   51          8E           0E   7B 000BD          EDIV    #14, (SP)+, R1, R1                           ;
                51          56           51   C3 000C2          SUBL3   R1, R6, R1                                   ; 1797
                6E          51           0E   A1 000C6          ADDW3   #14, R1, RDSC                                ;
                            51           6E   3C 000CA          MOVZWL  RDSC, R1                                     ; 1799
                            60           51   C0 000CD          ADDL2   R1, (R0)                                     ;
```

```
                                       0A  11 000D0         BRB     12$                                       ; 1785
              6E           56          02  A1 000D2 11$:    ADDW3   #2, R6, RDSC                              ; 1808
                     96    AB          03  88 000D6         BISB2   #3, -106(CCB)                             ; 1809
                                       60  D4 000DA         CLRL    (R0)                                      ; 1810
                     34    AB          6E  80 000DC 12$:    ADDB2   RDSC, 52(CCB)                             ; 1819
                     03    AE          01  90 000E0         MOVB    #1, RDSC+3                                ; 1820
                           50    0C    AC  D0 000E4         MOVL    ELEM_ADR, R0                              ; 1821
      6E      20     04    B0          56  2C 000E8         MOVC5   R6, @4(R0), #32, RDSC, @RDSC+4            ;
                                 04    BE     000EE                                                           ;
                           03          57  D1 000F0         CMPL    R7, #3                                    ; 1822
                                       0A  12 000F3         BNEQ    13$                                       ;
              50           56    04    AE  C1 000F5         ADDL3   RDSC+4, R6, R0                            ; 1824
                           60  0A0D    8F  B0 000FA         MOVW    #2573, (R0)                               ;
                                     01DE  31 000FF 13$:    BRW     47$                                       ; 1826
                                       59  DD 00102 14$:    PUSHL   CHARCONS                                  ; 1863
                           52    04    AC  D0 00104         MOVL    ELEM_TYPE, R2                             ;
                                       52  DD 00108         PUSHL   R2                                        ;
              16           06          52  CF 0010A         CASEL   R2, #6, #22                               ; 1848
  002E       0032         0032       0032     0010E 15$:    .WORD   17$-15$,-                                 ;
  002E       002E         0032       0032     00116                 17$-15$,-                                 ;
  002E       002E         002E       0038     0011E                 17$-15$,-                                 ;
  0038       002E         002E       002E     00126                 16$-15$,-                                 ;
  002E       002E         002E       002E     0012E                 17$-15$,-                                 ;
             0032         0032       002E     00136                 17$-15$,-                                 ;
                                                                    16$-15$,-                                 ;
                                                                    16$-15$,-                                 ;
                                                                    18$-15$,-                                 ;
                                                                    16$-15$,-                                 ;
                                                                    16$-15$,-                                 ;
                                                                    16$-15$,-                                 ;
                                                                    16$-15$,-                                 ;
                                                                    16$-15$,-                                 ;
                                                                    16$-15$,-                                 ;
                                                                    18$-15$,-                                 ;
                                                                    16$-15$,-                                 ;
                                                                    16$-15$,-                                 ;
                                                                    16$-15$,-                                 ;
                                                                    16$-15$,-                                 ;
                                                                    16$-15$,-                                 ;
                                                                    17$-15$,-                                 ;
                                                                    17$-15$                                   ;
                                       7E  D4 0013C 16$:    CLRL    -(SP)                                     ;
                                       0C  11 0013E         BRB     20$                                       ;
                           50    24    AE  9E 00140 17$:    MOVAB   D_VALUE, R0                               ;
                                       04  11 00144         BRB     19$                                       ;
                           50    1C    AE  9E 00146 18$:    MOVAB   DSC, R0                                   ;
                                       50  DD 0014A 19$:    PUSHL   R0                                        ;
                    0000V  CF          03  FB 0014C 20$:    CALLS   #3, GETFIELD                              ;
                           03          50  E9 00151         BLBC    R0, 21$                                   ;
                                     0087  31 00154         BRW     30$                                       ;
                                 FE    AB  95 00157 21$:    TSTB    -2(CCB)                                   ; 1870
                                       17  18 0015A         BGEQ    22$                                       ;
              12     A1    AB          04  E0 0015C         BBS     #4, -95(CCB), 22$                         ;
                           6E          02  D0 00161         MOVL    #2, TDSC                                  ; 1881
                     04    AE  FE94    CF  9E 00164         MOVAB   D_PROMPT, TDSC+4                          ; 1882
                                       5E  DD 0016A         PUSHL   SP                                        ; 1883
                00000000G  00          01  FB 0016C         CALLS   #1, BAS$OUT_T_DX_S                        ;
```

```
                              50      FF71   CB  9A 00173 22$:    MOVZBL  -143(CCB), R0                                   ; 1897
                              50 00000000G0040   D0 00178         MOVL    BAS$$AA_REC_PR1-104[R0], R0                     ;
                                 00000000G0040   16 00180         JSB     BAS$$AA_REC_PR1[R0]                             ;
                              06             50  E8 00187         BLBS    R0, 23$                                         ;
                                      34     AB  94 0018A         CLRB    52(CCB)                                         ; 1908
                                           0154  31 0018D         BRW     48$                                             ; 1909
                                      34     AB  94 00190 23$:    CLRB    52(CCB)                                         ; 1915
                                    0204     8F  BB 00193         PUSHR   #^M<R2,R9>                                      ; 1939
                16            06             52  CF 00197         CASEL   R2, #6, #22                                     ; 1924
      002E      0032          0032         0032     0019B 24$:    .WORD   26$-24$,-                                       ;
      002E      002E          0032         0032     001A3                 26$-24$,-                                       ;
      002E      002E          002E         0038     001AB                 26$-24$,-                                       ;
      0038      002E          002E         002E     001B3                 25$-24$,-                                       ;
      002E      002E          002E         002E     001BB                 26$-24$,-                                       ;
                0032          0032         002E     001C3                 26$-24$,-                                       ;
                                                                          25$-24$,-                                       ;
                                                                          25$-24$,-                                       ;
                                                                          27$-24$,-                                       ;
                                                                          25$-24$,-                                       ;
                                                                          25$-24$,-                                       ;
                                                                          25$-24$,-                                       ;
                                                                          25$-24$,-                                       ;
                                                                          25$-24$,-                                       ;
                                                                          25$-24$,-                                       ;
                                                                          27$-24$,-                                       ;
                                                                          25$-24$,-                                       ;
                                                                          25$-24$,-                                       ;
                                                                          25$-24$,-                                       ;
                                                                          25$-24$,-                                       ;
                                                                          25$-24$,-                                       ;
                                                                          26$-24$,-                                       ;
                                                                          26$-24$                                         ;
                                             7E  D4 001C9 25$:    CLRL    -(SP)                                           ;
                                             0C  11 001CB         BRB     29$                                             ;
                              50      24     AE  9E 001CD 26$:    MOVAB   D_VALUE, R0                                     ;
                                             04  11 001D1         BRB     28$                                             ;
                              50      1C     AE  9E 001D3 27$:    MOVAB   DSC, R0                                         ;
                                             50  DD 001D7 28$:    PUSHL   R0                                              ;
                        0000V CF             03  FB 001D9 29$:    CALLS   #3, GETFIELD                                    ;
                16            06             52  CF 001DE 30$:    CASEL   R2, #6, #22                                     ; 1947
      00CE      003E          0037         0030     001E2 31$:    .WORD   32$-31$,-                                       ;
      00CE      00CE          0045         003E     001EA                 33$-31$,-                                       ;
      00CE      00CE          00CE         005E     001F2                 34$-31$,-                                       ;
      0084      00CE          00CE         00CE     001FA                 45$-31$,-                                       ;
      00CE      00CE          00CE         00CE     00202                 34$-31$,-                                       ;
                004F          0045         00CE     0020A                 35$-31$,-                                       ;
                                                                          45$-31$,-                                       ;
                                                                          45$-31$,-                                       ;
                                                                          38$-31$,-                                       ;
                                                                          45$-31$,-                                       ;
                                                                          45$-31$,-                                       ;
                                                                          45$-31$,-                                       ;
                                                                          45$-31$,-                                       ;
                                                                          45$-31$,-                                       ;
                                                                          45$-31$,-                                       ;
                                                                          39$-31$,-                                       ;
                                                                          45$-31$,-                                       ;
```

```
                                                                      45$-31$,-                       :
                                                                      45$-31$,-                       :
                                                                      45$-31$,-                       :
                                                                      45$-31$,-                       :
                                                                      35$-31$,-                       :
                                                                      36$-31$                         :
                            2C   11 00210          BRB      37$                                       :
        0C    BC       1C   AE   90 00212 32$:     MOVB     D_VALUE, @ELEM_ADR                        : 1962
                            25   11 00217          BRB      37$                                       : 1947
        0C    BC       1C   AE   B0 00219 33$:     MOVW     D_VALUE, @ELEM_ADR                        : 1971
                            1E   11 0021E          BRB      37$                                       : 1947
        0C    BC       1C   AE   D0 00220 34$:     MOVL     D_VALUE, @ELEM_ADR                        : 1977
                            17   11 00225          BRB      37$                                       :
              50       0C   AC   D0 00227 35$:     MOVL     ELEM_ADR, R0                              : 1983
              60       1C   AE   7D 0022B          MOVQ     D_VALUE, (R0)                             :
                            7F   11 0022F          BRB      45$                                       : 1947
              50       0C   AC   D0 00231 36$:     MOVL     ELEM_ADR, R0                              : 1991
              60       1C   AE   7D 00235          MOVQ     D_VALUE, (R0)                             :
        08    A0       24   AE   7D 00239          MOVQ     D_VALUE+8, 8(R0)                          : 1993
                            70   11 0023E 37$:     BRB      45$                                       : 1947
        18    AE            59   D0 00240 38$:     MOVL     CHARCONS, DSC+4                           : 2001
        16    AE     010E   8F   B0 00244          MOVW     #270, DSC+2                               : 2003
                       14   AE   9F 0024A          PUSHAB   DSC                                       : 2005
                       0C   AC   DD 0024D          PUSHL    ELEM_ADR                                  :
   00000000G    00          02   FB 00250          CALLS    #2, STR$COPY_DX                           :
              1A       18   BE   91 00257          CMPB     @DSC+4, #26                               : 2006
                            53   12 0025B          BNEQ     45$                                       :
                       34   AB   94 0025D          CLRB     52(CCB)                                   : 2016
              7E      00G   8F   9A 00260          MOVZBL   #BAS$K_ENDFILDEV, -(SP)                   : 2017
                            47   11 00264          BRB      44$                                       :
        18    AE            59   D0 00266 39$:     MOVL     CHARCONS, DSC+4                           : 2033
        16    AE     010E   8F   B0 0026A          MOVW     #270, DSC+2                               : 2035
              50            5D   D0 00270          MOVL     FP, FMP                                   : 2041
              50       0C   A0   D0 00273 40$:     MOVL     12(FMP), FMP                              : 2045
              51 00000000G  00   9E 00277          MOVAB    BAS$HANDLER, R1                           : 2047
              51            60   D1 0027E          CMPL     (FMP), R1                                 :
                            04   13 00281          BEQL     41$                                       :
                            50   D5 00283          TSTL     FMP                                       : 2048
                            EC   12 00285          BNEQ     40$                                       :
                            50   D5 00287 41$:     TSTL     FMP                                       : 2050
                            09   13 00289          BEQL     42$                                       :
04         E6   A0          09   E1 0028B          BBC      #9, -26(FMP), 42$                         :
                            50   D4 00290          CLRL     FLAGS                                     : 2052
                            03   11 00292          BRB      43$                                       :
              50            08   D0 00294 42$:     MOVL     #8, FLAGS                                 : 2054
                            50   DD 00297 43$:     PUSHL    FLAGS                                     : 2056
                       0C   AC   DD 00299          PUSHL    ELEM_ADR                                  :
                       1C   AE   9F 0029C          PUSHAB   DSC                                       :
   00000000G    00          03   FB 0029F          CALLS    #3, BAS$CVT_T_P                           :
              07            50   E8 002A6          BLBS     STATUS, 45$                               : 2057
              7E      00G   8F   9A 002A9          MOVZBL   #BAS$K_DATFORERR, -(SP)                   :
              6A            01   FB 002AD 44$:     CALLS    #1, BAS$$STOP_IO                          :
                       34   AB   94 002B0 45$:     CLRB     52(CCB)                                   : 2060
              1B       0C   AB   B1 002B3          CMPW     12(CCB), #27                              : 2061
                            04   13 002B7          BEQL     46$                                       :
        FE    A8            04   8A 002B9          BICB2    #4, -2(TEMP_CCB)                          :
              50       2C   AE   9E 002BD 46$:     MOVAB    WORKSPACE, R0                             : 2065
```

```
            50              59  D1 002C1          CMPL    CHARCONS, R0                  ;
                            1A  13 002C4          BEQL    47$                           ;
                   0C       AE  9F 002C6          PUSHAB  UNWIND_VM_ADDR                ; 2068
                   14       AE  9F 002C9          PUSHAB  UNWIND_VM_SIZE                ;
   00000000G  00            02  FB 002CC          CALLS   #2, LIB$FREE_VM               ;
              0A            50  E8 002D3          BLBS    R0, 47$                       ;
                   10       AE  D4 002D6          CLRL    UNWIND_VM_SIZE                ; 2071
              7E   00G      8F  9A 002D9          MOVZBL  #BAS$K_PROLOSSOR, -(SP)       ; 2072
              6A            01  FB 002DD          CALLS   #1, BAS$$STOP_IO              ;
              50            01  D0 002E0 47$:     MOVL    #1, R0                        ; 2075
                                04 002E3          RET                                   ;
                            50  D4 002E4 48$:     CLRL    R0                            ; 2076
                                04 002E6          RET                                   ;
                              0000 002E7 49$:     .WORD   Save nothing                  ; 1689
              50   08       AC  D0 002E9          MOVL    8(AP), R0                     ;
              50   04       A0  D0 002ED          MOVL    4(R0), R0                     ;
                 FDDC       C0  9F 002F1          PUSHAB  UNWIND_CCB                    ;
                 FDE0       C0  9F 002F5          PUSHAB  UNWIND_VM_ADDR                ;
                 FDE4       C0  9F 002F9          PUSHAB  UNWIND_VM_SIZE                ;
                            03  DD 002FD          PUSHL   #3                            ;
                            5E  DD 002FF          PUSHL   SP                            ;
              7E   04       AC  7D 00301          MOVQ    4(AP), -(SP)                  ;
        0000V CF            03  FB 00305          CALLS   #3, UDF_RL1_HANDLER           ;
                                04 0030A          RET                                   ;

; Routine Size:  779 bytes,    Routine Base:  _BAS$CODE + 003C


;  784         2077  1
```

```
1 ROUTINE UDF_RL1_HANDLER (                              !Handeler for bas$udf_rl1
1         SIG                                           !Signal vector
1         ,MECH                                         !Mechanism vector
1         ,ENBL                                         !Enable vector
1         ) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       If we are unwinding and we have given the parsing space VM then
1 !       free this VM.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       SIG.rl.ra       A counted vector of parameters from LIB$SIGNAL/STOP
1 !       MECH.rl.ra      A counted vector of info from chf
1 !       ENBL.rl.ra      A counted vector of ENABLE argument addresses.
1 !
1 ! IMPLICIT INPUTS
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS
1 !
1 !       NONE
1 !
1 ! COMPLETION CODES
1 !
1 !       Always SS$_RESIGNAL, which is ignored when unwinding.
1 !
1 ! SIDE EFFECTS
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         SIG : REF VECTOR,
2         MECH: REF VECTOR,
2         ENBL: REF VECTOR;
2
2     GLOBAL REGISTER CCB = K_CCB_REG : REF BLOCK [,BYTE];
2
2     CCB = ..ENBL [3];
2 !+
2 ! If we are unwinding and have allocated VM then free it.
2 !-
3     IF (LIB$MATCH_COND ( SIG [1] , %REF(SS$_UNWIND) ) AND ( ..ENBL [1] GTRU 0 ))
2     THEN
2         IF NOT LIB$FREE_VM ( .ENBL [1] , .ENBL [2] )
2         THEN BAS$$STOP_IO ( BAS$K_PROLOSSOR );
2     RETURN (SS$_RESIGNAL);
2
1     END;
```

```
                                           0804 00000 UDF_RL1_HANDLER:
                                                              .WORD    Save R2,R11                          ; 2078
                    52        0C    AC   D0 00002             MOVL     ENBL, R2                             ; 2123
                    5B        0C    B2   D0 00006             MOVL     @12(R2), CCB                         ;
                    7E      0920    8F   3C 0000A             MOVZWL   #2336, -(SP)                         ; 2127
                                    5E   DD 0000F             PUSHL    SP                                   ;
        7E        04        AC           04   C1 00011        ADDL3    #4, SIG, -(SP)                       ;
          00000000G         00           02   FB 00016        CALLS    #2, LIB$MATCH_COND                   ;
                            1E           50   E9 0001D        BLBC     R0, 1$                               ;
                              04    B2   D5 00020             TSTL     @4(R2)                               ;
                                    19   13 00023             BEQL     1$                                   ;
                    7E        04    A2   7D 00025             MOVQ     4(R2), -(SP)                         ; 2129
          00000000G         00           02   FB 00029        CALLS    #2, LIB$FREE_VM                      ;
                            0B           50   E8 00030        BLBS     R0, 1$                               ;
                    7E       00G    8F   9A 00033             MOVZBL   #BAS$K_PROLOSSOR, -(SP)              ; 2130
          00000000G         00           01   FB 00037        CALLS    #1, BAS$$STOP_IO                     ;
                    50      0918    8F   3C 0003E 1$:         MOVZWL   #2328, R0                            ; 2131
                                         04 00043             RET                                           ; 2133

; Routine Size:  68 bytes,    Routine Base:  _BAS$CODE + 0347
;  1 GLOBAL ROUTINE BAS$$UDF_RL9
;  1         : JSB_UDF9 NOVALUE =
;  1
;  1 !++
;  1 ! FUNCTIONAL DESCRIPTION:
;  1 !
;  1 !       List directed input UDF termination.
;  1 !
;  1 ! FORMAL PARAMETERS:
;  1 !
;  1 !       NONE
;  1 !
;  1 ! IMPLICIT INPUTS:
;  1 !
;  1 !       NONE
;  1 !
;  1 ! IMPLICIT OUTPUTS:
;  1 !
;  1 !       NONE
;  1 !
;  1 ! ROUTINE VALUE:
;  1 ! COMPLETION CODES:
;  1 !
;  1 !       NONE
;  1 !
;  1 ! SIDE EFFECTS:
;  1 !
;  1 !       NONE
;  1 !
;  1 !--
;  1
;  2     BEGIN
;  2
;  2     RETURN;
;  1     END;



                                        05 00000 BAS$$UDF_RL9::
                                                  RSB                                              ; 2168

; Routine Size:  1 bytes,    Routine Base:  _BAS$CODE + 038B
```

```
1 ROUTINE GETFIELD (
1         ELEM,
1         ELEM_TYPE,
1         WORK_STR
1         ) :CALL_CCB =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Parse out the next input data field based on the field terminators
1 !       appropriate for the data type.  Return the field with tabs and spaces
1 !       stripped out in the area supplied by the calling routine.
1 !       A one is returned if a field was found.  A zero is returned if an <eol>
1 !       is encountered before a field is found.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       ELEM_TYPE.rlu.v         Type of element from list
1 !       ELEM.wz.r               Pointer of where to return the value
1 !                               May be a reference to a quadword or a descriptor
1 !       WORK_STR.wt.rs          Work string for parsing input string and resulting
1 !                               string for type text.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       LUB$A_BUF_PTR           current location in the buffer
1 !       LUB$A_BUF_END           pointer to last byte of buffer + 1
1 !       RAB$W_RSZ               buffer size
1 !       RAB$W_STV0              first word of STV field
1 !       RAB$W_STV2              second word of STV field
1 !       ISB$B_STTM_TYPE         I/O statement type in ISB
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       LUB$A_BUF_PTR           Pointer to next byte in user buffer
1 !       ISB$B_ERR_NO            first error found processing an I/O stmt.
1 !
1 ! ROUTINE VALUE:
1 !
1 !       1 = a data field was found
1 !       0 = a data field was not found
1 !
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
1     !+
1     ! Note: There are 3 exit points from this routine; not the best structure
1     ! but that's the way it is.
1     !-
1
```

```
        BEGIN

        MAP
            WORK_STR: REF VECTOR [K_WORK_STR_LEN, BYTE],    ! work area
            ELEM: REF VECTOR [2];                   ! default ELEM to numeric

        LOCAL
            DSC: BLOCK [8, BYTE],                   ! working desc. for parsing the input stream
            MASK,                                   ! mask value for SCANC
            RET_VAL,                                ! value to return to caller
            LEN,                                    ! length of input record remaining to be scanned
            SCAN_VAL,                               ! return value from SCANC
            PTRS,                                   ! char pointer for source string
            PTRD;                                   ! char pointer for destination string

        !+
        ! mask values for the SCANC to stop on different characters
        !-

        LITERAL
            K_COMMA = %X'01',
            K_SEMI = %X'02',
            K_SGL_QUOTE = %X'04',
            K_DBL_QUOTE = %X'08',
            K_TAB_SPACE = %X'10',
            K_NULL = %X'20',
            K_CHAR = %X'40',
            K_NONE = %X'00',
    !+
    ! The element size of a longword integer
    !-
            K_INT_SIZ = 4,
    !+
    ! The flags for floating and integer input
    !-
            K_INT_FLAGS = 5,
            K_FLT_F_FLAGS = 123,
            K_FLT_D_FLAGS = 115;

        BIND
            TABLE = UPLIT BYTE (
                %X'20', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40',
                    %X'10', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', ! column 0
                %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40',
                    %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', ! column 1
                %X'10', %X'40', %X'48', %X'40', %X'40', %X'40', %X'40', %X'44', %X'40',
                    %X'40', %X'40', %X'40', %X'41', %X'40', %X'40', %X'40', ! column 2
                %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40',
                    %X'40', %X'40', %X'42', %X'40', %X'40', %X'40', %X'40', ! column 3
                %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40',
                    %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', ! column 4
                %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40',
                    %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', ! column 5
                %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40',
                    %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', ! column 6
                %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40',
                    %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', ! column 7
            %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40',
                %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', ! column 8
            %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40',
                %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', ! column 9
            %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40',
                %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', ! column 10
            %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40',
                %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', ! column 11
            %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40',
                %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', ! column 12
            %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40',
                %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', ! column 13
            %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40',
                %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', ! column 14
            %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40',
                %X'40', %X'40', %X'40', %X'40', %X'40', %X'40', %X'40' ! column 15
        ): VECTOR[256, BYTE];

    EXTERNAL REGISTER
        CCB: REF BLOCK [, BYTE];

    !+
    ! Initialize the default null string (zero length)
    !-

    DSC[DSC$W_LENGTH] = 0;

    !+
    ! Check to see if there is any more data in the record.
    ! If there is no more data (BUF_PTR GEQA BUF_END) then return a failure
    ! status.  Otherwise, increment BUF_PTR.
    !-

    IF .CCB[LUB$A_BUF_PTR] GEQA .CCB[LUB$A_BUF_END]
    THEN
        RETURN 0
    ELSE
        CCB[LUB$A_BUF_PTR] = .CCB[LUB$A_BUF_PTR] + 1;

    !+
    ! Check for the buffer pointer equal to the end of the buffer (return default).
    ! If the statement type is INPUT LINE, we will do all of the other processing.
    ! For ANSI INPUT, no defaults should be applied.  Signal the 'too little data'
    ! error for ANSI.
    !-

    IF (.CCB [LUB$A_BUF_PTR] EQLA .CCB [LUB$A_BUF_END])
        AND .CCB [LUB$V_ANSI]
    THEN
        BAS$$SIGNAL_IO (BAS$K_TOOLITDAT);

    IF (.CCB[LUB$A_BUF_PTR] EQLA .CCB[LUB$A_BUF_END])
     AND (.CCB [ISB$B_STTM_TYPE] NEQ ISB$K_ST_TY_INL)
    THEN

        !+
        ! Return a zero or a null string as a default value
```

```
            !-

            BEGIN
            CASE .ELEM_TYPE
            FROM DSC$K_DTYPE_B TO DSC$K_DTYPE_H OF
            SET
            [INRANGE, OUTRANGE]:
                !+
                ! Data types not yet supported
                !-
                ELEM[0] = 0;
            [DSC$K_DTYPE_B, DSC$K_DTYPE_W, DSC$K_DTYPE_L, DSC$K_DTYPE_F]:

                !+
                ! Data type integer
                !-

                ELEM[0] = 0;
            [DSC$K_DTYPE_D, DSC$K_DTYPE_G]:

                !+
                ! Data type double precision or g float
                !-

                BEGIN
                ELEM[0] = 0;
                ELEM[1] = 0;
                END;
            [DSC$K_DTYPE_H]:

                !+
                ! Data type h float
                !-
                BEGIN
                ELEM[0] = 0;
                ELEM[1] = 0;
                ELEM[2] = 0;
                ELEM[3] = 0;
                END;
            [DSC$K_DTYPE_T, DSC$K_DTYPE_P]:

                !+
                ! Data type text or packed decimal string
                !-
                BEGIN
                MAP
                    ELEM: REF BLOCK [8, BYTE];
                ELEM[DSC$W_LENGTH] = 0;
                END;
            TES;
            RETURN 1;
            END;

        !+
        ! Set up the mask for the scan.  Make any special adjustments to the buffer
        ! pointer that are necessary for type character string.
        !-
```

```

DSC[DSC$A_POINTER] = WORK_STR[0];
CASE .ELEM_TYPE
FROM DSC$K_DTYPE_B TO DSC$K_DTYPE_H OF
SET
[INRANGE, OUTRANGE]:
!+
! Data types which are not supported yet
!-
0;
[DSC$K_DTYPE_B, DSC$K_DTYPE_W, DSC$K_DTYPE_L, DSC$K_DTYPE_F, DSC$K_DTYPE_D,
 DSC$K_DTYPE_G, DSC$K_DTYPE_H, DSC$K_DTYPE_P]:
    MASK = K_COMMA OR K_TAB_SPACE OR K_NULL;
[DSC$K_DTYPE_T]:

    !+
    ! First check for INPUT LINE, MAT LINPUT, or LINPUT.  They return the whole line regardless
    ! of the contents.  Remove all leading tabs and spaces.  Next check for
    ! quotes (single or double).  They return
    ! everything up to the matched quote.  The quotes themselves are not returned
    ! and the first one is stripped off by incrementing the buffer pointer.
    ! Otherwise, a field is delimited by a comma or <eol>.
    ! Trailing spaces and tabs are stripped off unquoted strings at great
    ! pain.
    !-

    IF .CCB[ISB$B_STTM_TYPE] EQL ISB$K_ST_TY_LIN
     OR .CCB[ISB$B_STTM_TYPE] EQL ISB$K_ST_TY_INL
     OR .CCB [ISB$B_STTM_TYPE] EQL ISB$K_ST_TY_MLI
    THEN
        MASK = K_NONE
    ELSE
        BEGIN

        !+
        ! Strip off the leading tabs, nulls, and spaces.  If this results
        ! in a zero length string then return the null string.
        !-

        WHILE (.(.CCB [LUB$A_BUF_PTR])<0,8,0> EQL %C' '
         OR .(.CCB [LUB$A_BUF_PTR])<0,8,0> EQL %C'	'
         OR .(.CCB [LUB$A_BUF_PTR])<0,8,0> EQL %X'00')
         AND .CCB [LUB$A_BUF_PTR] LSS .CCB [LUB$A_BUF_END]
        DO
            CCB[LUB$A_BUF_PTR] = .CCB[LUB$A_BUF_PTR] + 1;
        IF .CCB[LUB$A_BUF_PTR] GEQ .CCB[LUB$A_BUF_END]
         OR .(.CCB[LUB$A_BUF_PTR])<0,8,0> EQL %C','
        THEN
            BEGIN
            MAP
                ELEM: REF BLOCK [8, BYTE];
            ELEM[DSC$W_LENGTH] = 0;
            RETURN 1;
            END;
        IF .(.CCB[LUB$A_BUF_PTR])<0, 8> EQL %C''''
        THEN
            BEGIN
```

```
4                   MASK = K_SGL_QUOTE;
4                   CCB[LUB$A_BUF_PTR] = .CCB[LUB$A_BUF_PTR] + 1;
4                   END
3               ELSE
3                   IF .(.CCB[LUB$A_BUF_PTR])<0, 8> EQL %C'"'
3                   THEN
4                       BEGIN
4                       MASK = K_DBL_QUOTE;
4                       CCB[LUB$A_BUF_PTR] = .CCB[LUB$A_BUF_PTR] + 1;
4                       END
3                   ELSE
3                       MASK = K_COMMA;
2               END;
2       TES;
2
2       !+
2       ! Point the character pointer to the start of the field.
2       !-
2
2       PTRS = CH$PTR(.CCB[LUB$A_BUF_PTR]);
2       PTRD = CH$PTR(.DSC[DSC$A_POINTER]);
2       LEN = .CCB[LUB$A_BUF_END] - .CCB[LUB$A_BUF_PTR];
2
2       !+
2       ! Based on the data type, scan the input data string for an element
2       !-
2
2       WHILE 1 DO
3           BEGIN
3           LITERAL
3               K_DECIMAL_PT = %X'2E';
3           LOCAL
3               TEMP_LEN;                       !Used to allow > 64kb data
3           TEMP_LEN = (IF .LEN GEQU 65536 THEN 65535 ELSE .LEN);
3           SCAN_VAL = SCANC(TEMP_LEN, .CCB[LUB$A_BUF_PTR], TABLE, MASK);
3           IF .SCAN_VAL NEQ 0
3           THEN
3               CASE .ELEM_TYPE
3               FROM DSC$K_DTYPE_B TO DSC$K_DTYPE_H OF
3               SET
3               [DSC$K_DTYPE_B, DSC$K_DTYPE_W, DSC$K_DTYPE_L, DSC$K_DTYPE_F,
3                DSC$K_DTYPE_D, DSC$K_DTYPE_G, DSC$K_DTYPE_H, DSC$K_DTYPE_P]:
4                   BEGIN
4                   CH$MOVE (.SCAN_VAL-.CCB[LUB$A_BUF_PTR], .PTRS, .PTRD);
5                   IF      (.(.SCAN_VAL)<0, 8> EQL K_TAB)
5                       OR (.(.SCAN_VAL)<0, 8> EQL K_SP)
5                       OR (.(.SCAN_VAL)<0, 8> EQL %X'00')
4                   THEN
4
4                       !+
4                       ! A tab, null, or a space has been found in a numeric field
4                       ! Strip it out.
4                       ! Also strips out decimal points for packed decimal.
4                       !-
4
5                       BEGIN
5                       DSC[DSC$W_LENGTH] = .DSC[DSC$W_LENGTH] + (.SCAN_VAL - .CCB[LUB$A_BUF_PTR]);
```

```
5                       PTRS = CH$PLUS(.PTRS, .SCAN_VAL-.CCB[LUB$A_BUF_PTR] + 1);
5                       PTRD = CH$PLUS(.PTRD, .SCAN_VAL - .CCB[LUB$A_BUF_PTR]);
5                       LEN = .LEN - (.SCAN_VAL - .CCB[LUB$A_BUF_PTR]) - 1;
5                       CCB[LUB$A_BUF_PTR] = .SCAN_VAL + 1;
5                       END
4                   ELSE
5                       BEGIN
5                       IF .SCAN_VAL EQLU .CCB[LUB$A_BUF_PTR]
5                       THEN
5
5                           !+
5                           ! An element separator was encountered as the next character;
5                           ! return the proper default value or the data scanned so far.
5                           !-
6                           BEGIN
6                           RET_VAL = 1;
6                           EXITLOOP;
5                           END;
5                       DSC[DSC$W_LENGTH] = .DSC[DSC$W_LENGTH] + .SCAN_VAL - .CCB[LUB$A_BUF_PTR];
5                       LEN = .LEN - (.SCAN_VAL - .CCB[LUB$A_BUF_PTR]) - 1;
5                       CCB[LUB$A_BUF_PTR] = .SCAN_VAL;
5                       RET_VAL = 1;
5                       EXITLOOP;
4                       END;
3                   END;
3
3           [DSC$K_DTYPE_T]:
3               !+
3               ! Type text
3               ! Update the length so far, move the substring found, and
3               ! check for a delimiting quote if necessary.
3               !-
3
4               BEGIN
4               LOCAL
4                   A_HIGH_MARK;                    ! High water mark of SCAN
4
4               !+
4               ! Strip off trailing spaces, nulls, and tabs if unquoted string
4               !-
4
4               A_HIGH_MARK = .SCAN_VAL;
4               IF .MASK EQL K_COMMA
4               THEN
4                   WHILE .(.SCAN_VAL - 1)<0,8,0> EQL %C' '
4                     OR .(.SCAN_VAL - 1)<0,8,0> EQL %C'	'
4                     OR .(.SCAN_VAL - 1)<0,8,0> EQL %X'00'
4                   DO
4                       SCAN_VAL = .SCAN_VAL - 1;
4
4               DSC[DSC$W_LENGTH] = .SCAN_VAL - .CCB[LUB$A_BUF_PTR];
4               CH$MOVE (.SCAN_VAL - .CCB[LUB$A_BUF_PTR], .PTRS, .PTRD);
4
4               !+
4               ! increment the buffer pointer if a delimiting quote is present
4               !-
4
```

```
                CCB[LUB$A_BUF_PTR] = .A_HIGH_MARK;
                IF .(.A_HIGH_MARK)<0, 8> EQL %C'''' OR .(.A_HIGH_MARK)<0, 8> EQL %C'"'
                THEN
                    BEGIN
                    LOCAL
                        T_RET_VAL,                      ! temp return value from SCANC
                                                        ! looking for delimiting comma
                        REM_LENGTH;                     ! Length remaining in the buffer
                    CCB[LUB$A_BUF_PTR] = .CCB[LUB$A_BUF_PTR] + 1;
!+
! Scan for a comma, another character or the end-of-record following this quoted string.
! Set BUF_PTR to the address that the scan returns.  If there is a comma,
! then it will be pointing at the comma.
! If there is a character other than space, tab or null following qoute, signal.
!-
                    MASK = K_COMMA OR K_CHAR;
                    REM_LENGTH = .LEN - .DSC [DSC$W_LENGTH] - 1;
                    REM_LENGTH = (IF .REM_LENGTH GEQU 65536 THEN 65535 ELSE .REM_LENGTH);
                    T_RET_VAL = SCANC(REM_LENGTH, .CCB [LUB$A_BUF_PTR],
                        TABLE, MASK);
                    CCB [LUB$A_BUF_PTR] = (IF .T_RET_VAL EQL 0 THEN .CCB [LUB$A_BUF_END] + 1 ELSE .T_RET_VAL
                    IF (.T_RET_VAL NEQ 0) AND
                       (.(.T_RET_VAL)< 0 , 8 > NEQ %C',')
                    THEN BAS$$STOP_IO ( BAS$K_DATFORERR );
                    END;
                RET_VAL = 1;
                EXITLOOP;
                END;
            [INRANGE, OUTRANGE]:
                !+
                ! Data types which are not supported
                !-
                0;
            TES
        ELSE

            !+
            ! The whole rest of the buffer was scanned without finding an element separator
            !-

            BEGIN
            LOCAL
                T_BUF_END;                              ! temp to hold BUF_END for deleting
                                                        ! trailing nulls, spaces, and tabs
            T_BUF_END = .CCB[LUB$A_BUF_END];

            !+
            ! Check the mask value and if it indicates that this string is
            ! bound by quotes, then check to see if LUB$A_BUF_PTR is not
            ! equal to LUB$A_BUF_END.  The assumption is that if BUF_PTR is
            ! equal to BUF_END, then a delimiting quote was not found but
            ! rather the SCANC stopped on end-of-record.
            !-

            IF .MASK EQL K_DBL_QUOTE OR .MASK EQL K_SGL_QUOTE
            THEN
                BAS$$STOP_IO(BAS$K_DATFORERR);
```

```
4
4               !+
4               ! So far everything is OK.  Move the data, then check for INPUT LINE
4               ! If this is an INPUT LINE, then we need to bump the length based on
4               ! the terminator and move the terminator into the buffer.
4               ! If INPUT then strip off the trailing spaces, nulls, and tabs
4               !-
4
5               IF (.CCB[ISB$B_STTM_TYPE] EQL ISB$K_ST_TY_INP
5                OR .CCB[ISB$B_STTM_TYPE] EQL ISB$K_ST_TY_REA)
4                AND .ELEM_TYPE EQL DSC$K_DTYPE_T
4               THEN
4                   WHILE .(.T_BUF_END - 1)<0,8,0> EQL %C' '
4                    OR .(.T_BUF_END - 1)<0,8,0> EQL %C'	'
4                    OR .(.T_BUF_END - 1)<0,8,0> EQL %X'00'
4                   DO
4                       T_BUF_END = .T_BUF_END - 1;
4
4               DSC[DSC$W_LENGTH] = .DSC[DSC$W_LENGTH] + (.T_BUF_END - .CCB[LUB$A_BUF_PTR]);
4               PTRD = CH$MOVE (.T_BUF_END - .CCB[LUB$A_BUF_PTR], .PTRS, .PTRD);
4               IF .CCB[ISB$B_STTM_TYPE] EQL ISB$K_ST_TY_INL
4               THEN
4
4                   !+
4                   ! This is an INPUT LINE.  Bump length and tack on the terminator
4                   !-
4
5                   BEGIN
5                   LITERAL
5                       K_ESCAPE = %X'1B',              ! ASCII escape character
5                       K_CR = %X'0D',                  ! ASCII carriage return char.
5                       K_CRLF = %X'0A0D';              ! ASCII carriage return-line
5                                                       ! feed char. combination
5 !+
5 ! Due to an undocumented change to RMS for V2.0, we want to look only at the
5 ! low order byte to find the terminating character.  RMS is now returning the
5 ! length of the terminating sequence in the upper word.
5 !-
5                   SELECTONEU .CCB [RAB$W_STV0] OF
5                   SET
5                   [K_ESCAPE]:
6                       BEGIN
6 !+
6 ! Check to see if the length is one.  If it is, we have to move the escape
6 ! character by hand; it is not at the end of the buffer.  Otherwise, the escape
6 ! sequence is at the end of the buffer following the data.
6 !-
6                       IF .CCB [RAB$W_STV2] EQLU 1
6                       THEN
7                           BEGIN
7                           DSC [DSC$W_LENGTH] = .DSC [DSC$W_LENGTH] + 1;
7                           CH$MOVE(1, UPLIT(K_ESCAPE), .PTRD);
7                           END
6                       ELSE
7                           BEGIN
7                           DSC[DSC$W_LENGTH] = .DSC[DSC$W_LENGTH] + .CCB [RAB$W_STV2];
7                           CH$MOVE (.CCB [RAB$W_STV2], .CCB [RAB$L_RBF] + .CCB [RAB$W_RSZ], .PTRD);
```

```
6                              END;
5                          END;
5                      [K_CR]:
6                          BEGIN
6                          DSC[DSC$W_LENGTH] = .DSC[DSC$W_LENGTH] + 2;
6                          CH$MOVE (2, UPLIT(K_CRLF), .PTRD);
5                          END;
5                      [OTHERWISE]:
5                          ;
5                      TES;
4                      END;
4                  CCB[LUB$A_BUF_PTR] = .CCB[LUB$A_BUF_END];
4                  RET_VAL = 1;
4                  EXITLOOP;
3                  END;
2              END;            ! WHILE loop
2  !+
2  ! Update the data pointer if this is a READ or MAT READ so that we are pointing
2  ! at the next data element in the event of an error.
2  !-
2          IF (.CCB [ISB$B_STTM_TYPE] EQL ISB$K_ST_TY_MRE) OR
3           (.CCB [ISB$B_STTM_TYPE] EQL ISB$K_ST_TY_REA)
2          THEN
3              BEGIN
3              LOCAL
3                  BMF : REF BLOCK [0, BYTE] FIELD (BSF$MAJOR_FRAME);      ! BASIC major frame pointer
3              BMF = .CCB [ISB$A_MAJ_F_PTR];
3              BMF [BSF$A_CUR_DTA] = .CCB [LUB$A_BUF_PTR] + 1;
2              END;
2
2
2          !+
2          ! Convert the field that was found into internal format
2          !-
2
3          IF NOT (CASE .ELEM_TYPE
3              FROM DSC$K_DTYPE_B TO DSC$K_DTYPE_H OF
3              SET
3              [INRANGE, OUTRANGE]:
3                  !+
3                  ! Data types that are not yet supported
3                  !-
4                  BEGIN
4                  0
3                  END;
3              [DSC$K_DTYPE_B]:
3                  !+
3                  ! Integer - byte
3                  ! Do the conversion and then check the range.
3                  !-
3
4                  BEGIN
4                  IF OTS$CVT_TI_L(DSC, ELEM[0], K_INT_SIZ, K_INT_FLAGS)
4                  THEN
4
4                      !+
4                      ! The conversion was successful.
```

```
; 1449          2739  4                     !-
; 1450          2740  4
; 1451          2741  4                     IF .ELEM[0] GTR 127
; 1452          2742  4                      OR .ELEM[0] LSS -128
; 1453          2743  4                     THEN
; 1454          2744  4                         BAS$$STOP_IO (BAS$K_ILLNUM)
; 1455          2745  4                     ELSE
; 1456          2746  4                         1                              ! signify success
; 1457          2747  4                 ELSE
; 1458          2748  4
; 1459          2749  4                     !+
; 1460          2750  4                     ! The conversion routine returned failure.
; 1461          2751  4                     !-
; 1462          2752  4
; 1463          2753  4                     0
; 1464          2754  3                 END;
; 1465          2755  3             [DSC$K_DTYPE_W]:
; 1466          2756  3                 !+
; 1467          2757  3                 ! Integer - word
; 1468          2758  3                 ! Do the conversion of the value input and then range check
; 1469          2759  3                 ! for overflow.
; 1470          2760  3                 !-
; 1471          2761  3
; 1472          2762  4                 BEGIN
; 1473          2763  4                 IF OTS$CVT_TI_L(DSC, ELEM[0], K_INT_SIZ, K_INT_FLAGS)
; 1474          2764  4                 THEN
; 1475          2765  4
; 1476          2766  4                     !+
; 1477          2767  4                     ! The conversion was successful.  Check the range of the
; 1478          2768  4                     ! value input.  Signal an error or assume a value of success.
; 1479          2769  4                     !-
; 1480          2770  4
; 1481          2771  4                     IF .ELEM[0] GTR 32767
; 1482          2772  4                      OR .ELEM[0] LSS -32768
; 1483          2773  4                     THEN
; 1484          2774  4                         BAS$$STOP_IO (BAS$K_ILLNUM)
; 1485          2775  4                     ELSE
; 1486          2776  4                         1                    ! signify success
; 1487          2777  4                 ELSE
; 1488          2778  4
; 1489          2779  4                     !+
; 1490          2780  4                     ! The conversion routine returned failure.  Assume a value of
; 1491          2781  4                     ! failure.
; 1492          2782  4                     !-
; 1493          2783  4
; 1494          2784  4                     0
; 1495          2785  3                 END;
; 1496          2786  3             [DSC$K_DTYPE_L]:
; 1497          2787  3
; 1498          2788  3                 !+
; 1499          2789  3                 ! Integer - longword.  Upper and lower bounds checking is performed
; 1500          2790  3                 ! by the conversion routine.
; 1501          2791  3                 !-
; 1502          2792  3
; 1503          2793  4                 BEGIN
; 1504          2794  4                 OTS$CVT_TI_L(DSC, ELEM[0], K_INT_SIZ, K_INT_FLAGS)
; 1505          2795  3                 END;
```

```
3              [DSC$K_DTYPE_F]:
3                  !+
3                  ! floating single precision
3                  !-
4                  BEGIN
4                  LOCAL
4                      T_ELEM: VECTOR[2];  ! temp. quadword work area
4                  IF OTS$CVT_T_D(DSC, T_ELEM, 0, 0, K_FLT_F_FLAGS)
4                      THEN LIB$CVTDF(T_ELEM[0], ELEM[0])
4                      ELSE 0
3                  END;
3              [DSC$K_DTYPE_D]:
3                  !+
3                  ! double precision floating
3                  !-
3
4                      BEGIN
4                      LOCAL
4                          STATUS;
4                      STATUS = OTS$CVT_T_D(DSC, ELEM[0], 0, .CCB [ISB$B_SCALE_FAC], K_FLT_D_FLAGS);
4  !+
4  ! Truncate any fractional portion remaining if scaling is done.
4  !-
4                      IF .CCB [ISB$B_SCALE_FAC] NEQ 0
4                      THEN
5                          BEGIN
5                          MTH$DINT(ELEM [0]);
6                              BEGIN
6                              REGISTER
6                                  R0 = 0,
6                                  R1 = 1;
6                              ELEM [0] = .R0;
6                              ELEM [1] = .R1;
5                              END;
4                          END;
4                      .STATUS
3                      END;
3              [DSC$K_DTYPE_G]:
3                  !+
3                  ! g floating
3                  !-
4                  BEGIN
4                  LOCAL
4                      STATUS;
4                  STATUS = OTS$CVT_T_G(DSC, ELEM[0], 0, 0, K_FLT_D_FLAGS);
4                  .STATUS
3                  END;
3              [DSC$K_DTYPE_H]:
3                  !+
3                  ! h floating
3                  !-
4                  BEGIN
4                  LOCAL
4                      STATUS;
4                  STATUS = OTS$CVT_T_H(DSC, ELEM[0], 0, 0, K_FLT_D_FLAGS);
4                  .STATUS
3                  END;
```

```
; 1563          2853  3                 [DSC$K_DTYPE_T, DSC$K_DTYPE_P]:
; 1564          2854  3                     !+
; 1565          2855  3                     ! String or packed - no conversion - just return success
; 1566          2856  3                     !-
; 1567          2857  4                     BEGIN
; 1568          2858  4                     MAP
; 1569          2859  4                         ELEM: REF BLOCK [8, BYTE];
; 1570          2860  4                     ELEM[DSC$W_LENGTH] = .DSC[DSC$W_LENGTH];
; 1571          2861  4                     1
; 1572          2862  3                     END;
; 1573          2863  3                     TES)
; 1574          2864  2                 THEN
; 1575          2865  2                     BAS$$STOP_IO(BAS$K_DATFORERR);
; 1576          2866  2         RETURN .RET_VAL;
; 1577          2867  1         END;
; INFO#250              L1:2827
; Referenced REGISTER symbol R0 is probably not initialized
; INFO#250              L1:2828
; Referenced REGISTER symbol R1 is probably not initialized


40 40 40 40 40 10 40 40 40 40 40 40 40 40 20  0038C P.AAC:  .BYTE   32, 64, 64, 64, 64, 64, 64, 64, 64, 16, -    ;
40 40 40 40 40 40 40 40 40 40 40 40 40 40 40  0039B                 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, -    ;
41 40 40 40 40 44 40 40 40 40 48 40 10 40 40  003AA                 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, -    ;
42 40 40 40 40 40 40 40 40 40 40 40 40 40 40  003B9                 64, 64, 16, 64, 72, 64, 64, 64, 64, 68, -    ;
40 40 40 40 40 40 40 40 40 40 40 40 40 40 40  003C8                 64, 64, 64, 64, 65, 64, 64, 64, 64, 64, -    ;
40 40 40 40 40 40 40 40 40 40 40 40 40 40 40  003D7                 64, 64, 64, 64, 64, 64, 64, 64, 64, 66, -    ;
40 40 40 40 40 40 40 40 40 40 40 40 40 40 40  003E6                 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, -    ;
40 40 40 40 40 40 40 40 40 40 40 40 40 40 40  003F5                 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, -    ;
40 40 40 40 40 40 40 40 40 40 40 40 40 40 40  00404                 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, -    ;
40 40 40 40 40 40 40 40 40 40 40 40 40 40 40  00413                 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, -    ;
40 40 40 40 40 40 40 40 40 40 40 40 40 40 40  00422                 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, -    ;
40 40 40 40 40 40 40 40 40 40 40 40 40 40 40  00431                 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, -    ;
40 40 40 40 40 40 40 40 40 40 40 40 40 40 40  00440                 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, -    ;
40 40 40 40 40 40 40 40 40 40 40 40 40 40 40  0044F                 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, -    ;
40 40 40 40 40 40 40 40 40 40 40 40 40 40 40  0045E                 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, -    ;
40 40 40 40 40 40 40 40 40 40 40 40 40 40 40  0046D                 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, -    ;
40 40 40 40 40 40 40 40 40 40 40 40 40 40 40  0047C                 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, -    ;
                                          40  0048B                 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, -    ;
                                                                    64, 64, 64, 64, 64, 64, 64, 64, 64, 64, -    ;
                                                                    64, 64, 64, 64, 64, 64, 64, 64, 64, 64, -    ;
                                                                    64, 64, 64, 64, 64, 64, 64, 64, 64, 64, -    ;
                                                                    64, 64, 64, 64, 64, 64, 64, 64, 64, 64, -    ;
                                                                    64, 64, 64, 64, 64, 64, 64, 64, 64, 64, -    ;
                                                                    64, 64, 64, 64, 64, 64, 64, 64, 64, 64, -    ;
                                                                    64, 64, 64, 64, 64, 64, 64, 64, 64, 64, -    ;
                                                                    64, 64, 64, 64, 64, 64                       ;
                              0000001B  0048C P.AAD:  .LONG   27                                           ;
                              00000A0D  00490 P.AAE:  .LONG   2573                                         ;

                                              TABLE=                P.AAC

                                   07FC 00000 GETFIELD:
                                                      .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10             ; 2169
                    5E             20  C2 00002       SUBL2   #32, SP                                      ;
```

```
                                18   AE  B4 00005         CLRW    DSC                                   ; 2308
                    58          B0   AB  9E 00008         MOVAB   -80(CCB), R8                          ; 2316
                                B4   AB  9F 0000C         PUSHAB  -76(CCB)                              ;
                00  BE               68  D1 0000F         CMPL    (R8), @0(SP)                          ;
                                     03  1F 00013         BLSSU   1$                                    ;
                                   045F  31 00015         BRW     77$                                   ;
                                     68  D6 00018 1$:     INCL    (R8)                                  ; 2320
                                     52  D4 0001A         CLRL    R2                                    ; 2329
                00  BE               68  D1 0001C         CMPL    (R8), @0(SP)                          ;
                                     12  12 00020         BNEQ    2$                                    ;
                                     52  D6 00022         INCL    R2                                    ;
          0B       A1   AB           04  E1 00024         BBC     #4, -95(CCB), 2$                      ; 2330
                        7E      00G  8F  9A 00029         MOVZBL  #BAS$K_TOOLITDAT, -(SP)               ; 2332
             00000000G  00           01  FB 0002D         CALLS   #1, BAS$$SIGNAL_IO                    ;
                        52           52  E9 00034 2$:     BLBC    R2, 9$                                ; 2334
                        20     FF71  CB  91 00037         CMPB    -143(CCB), #32                        ; 2335
                                     4B  13 0003C         BEQL    9$                                    ;
                        50      04   AC  D0 0003E         MOVL    ELEM, R0                              ; 2350
          16            06      08   AC  CF 00042         CASEL   ELEM_TYPE, #6, #22                    ; 2343
 002E     002E        002E         002E     00047 3$:     .WORD   4$-3$,-                               ;
 002E     002E        0032         002E     0004F                 4$-3$,-                               ;
 002E     002E        002E         003D     00057                 4$-3$,-                               ;
 003D     002E        002E         002E     0005F                 4$-3$,-                               ;
 002E     002E        002E         002E     00067                 4$-3$,-                               ;
          0036        0032         002E     0006F                 5$-3$,-                               ;
                                                                  4$-3$,-                               ;
                                                                  4$-3$,-                               ;
                                                                  7$-3$,-                               ;
                                                                  4$-3$,-                               ;
                                                                  4$-3$,-                               ;
                                                                  4$-3$,-                               ;
                                                                  4$-3$,-                               ;
                                                                  4$-3$,-                               ;
                                                                  4$-3$,-                               ;
                                                                  7$-3$,-                               ;
                                                                  4$-3$,-                               ;
                                                                  4$-3$,-                               ;
                                                                  4$-3$,-                               ;
                                                                  4$-3$,-                               ;
                                                                  4$-3$,-                               ;
                                                                  5$-3$,-                               ;
                                                                  6$-3$                                 ;
                                     60  D4 00075 4$:     CLRL    (R0)                                  ; 2357
                                     0D  11 00077         BRB     8$                                    ;
                                     60  7C 00079 5$:     CLRQ    (R0)                                  ; 2365
                                     09  11 0007B         BRB     8$                                    ; 2343
                                     60  7C 0007D 6$:     CLRQ    (R0)                                  ; 2374
                                08   A0  7C 0007F         CLRQ    8(R0)                                 ; 2376
                                     02  11 00082         BRB     8$                                    ; 2343
                                     60  B4 00084 7$:     CLRW    (R0)                                  ; 2387
                                   0083  31 00086 8$:     BRW     18$                                   ; 2390
                  20    AE      0C   AC  D0 00089 9$:     MOVL    WORK_STR, DSC+4                       ; 2398
          16            06      08   AC  CF 0008E         CASEL   ELEM_TYPE, #6, #22                    ; 2399
 009B     0030        0030         0030     00093 10$:    .WORD   11$-10$,-                             ;
 009B     009B        0030         0030     0009B                 11$-10$,-                             ;
 009B     009B        009B         0036     000A3                 11$-10$,-                             ;
 0030     009B        009B         009B     000AB                 23$-10$,-                             ;
```

```
     009B          009B          009B          009B   000B3              11$-10$,-                          :
                   0030          0030          009B   000BB              11$-10$,-                          :
                                                                         23$-10$,-                          :
                                                                         23$-10$,-                          :
                                                                         12$-10$,-                          :
                                                                         23$-10$,-                          :
                                                                         23$-10$,-                          :
                                                                         23$-10$,-                          :
                                                                         23$-10$,-                          :
                                                                         23$-10$,-                          :
                                                                         23$-10$,-                          :
                                                                         11$-10$,-                          :
                                                                         23$-10$,-                          :
                                                                         23$-10$,-                          :
                                                                         23$-10$,-                          :
                                                                         23$-10$,-                          :
                                                                         23$-10$,-                          :
                                                                         11$-10$,-                          :
                                                                         11$-10$                            :
                                             6B  11 000C1          BRB   23$                                :
                             0C    AE        31  D0 000C3 11$:     MOVL  #49, MASK                          : 2409
                                             65  11 000C7          BRB   23$                                :
                                   50   FF71  CB  9A 000C9 12$:    MOVZBL -143(CCB), R0                     : 2423
                                   1C        50  91 000CE          CMPB  R0, #28                            :
                                             0A  13 000D1          BEQL  13$                                :
                                   20        50  91 000D3          CMPB  R0, #32                            : 2424
                                             05  13 000D6          BEQL  13$                                :
                                   32        50  91 000D8          CMPB  R0, #50                            : 2425
                                             05  12 000DB          BNEQ  14$                                :
                                        0C   AE  D4 000DD 13$:     CLRL  MASK                               : 2427
                                             4C  11 000E0          BRB   23$                                :
                                   20   00   B8  91 000E2 14$:     CMPB  @0(R8), #32                        : 2436
                                             0B  13 000E6          BEQL  15$                                :
                                   09   00   B8  91 000E8          CMPB  @0(R8), #9                         : 2437
                                             05  13 000EC          BEQL  15$                                :
                                        00   B8  95 000EE          TSTB  @0(R8)                             : 2438
                                             0A  12 000F1          BNEQ  16$                                :
                             00    BE        68  D1 000F3 15$:     CMPL  (R8), @0(SP)                       : 2439
                                             04  18 000F7          BGEQ  16$                                :
                                             68  D6 000F9          INCL  (R8)                               : 2441
                                             E5  11 000FB          BRB   14$                                :
                             00    BE        68  D1 000FD 16$:     CMPL  (R8), @0(SP)                       : 2442
                                             06  18 00101          BGEQ  17$                                :
                                   2C   00   B8  91 00103          CMPB  @0(R8), #44                        : 2443
                                             07  12 00107          BNEQ  19$                                :
                                        04   BC  B4 00109 17$:     CLRW  @ELEM                              : 2448
                                   50        01  D0 0010C 18$:     MOVL  #1, R0                             : 2449
                                                 04 0010F          RET                                      :
                                   27   00   B8  91 00110 19$:     CMPB  @0(R8), #39                        : 2451
                                             06  12 00114          BNEQ  20$                                :
                             0C    AE        04  D0 00116          MOVL  #4, MASK                           : 2454
                                             0A  11 0011A          BRB   21$                                : 2455
                                   22   00   B8  91 0011C 20$:     CMPB  @0(R8), #34                        : 2458
                                             08  12 00120          BNEQ  22$                                :
                             0C    AE        08  D0 00122          MOVL  #8, MASK                           : 2461
                                             68  D6 00126 21$:     INCL  (R8)                               : 2462
                                             04  11 00128          BRB   23$                                : 2458
```

```
                      0C   AE          01 D0 0012A 22$:   MOVL   #1, MASK                              ; 2465
                           5A          68 D0 0012E 23$:   MOVL   (R8), PTRS                            ; 2473
                      04   AE     20   AE D0 00131        MOVL   DSC+4, PTRD                           ; 2474
                 57   00   BE          68 C3 00136        SUBL3  (R8), @0(SP), LEN                     ; 2475
           00010000        8F          57 D1 0013B 24$:   CMPL   LEN, #65536                           ; 2487
                                       07 1F 00142        BLSSU  25$                                   ;
                           50   FFFF   8F 3C 00144        MOVZWL #65535, TEMP_LEN                      ;
                                       03 11 00149        BRB    26$                                   ;
                           50          57 D0 0014B 25$:   MOVL   LEN, TEMP_LEN                         ;
  0C   AE   FDA3   CF      00   B8     50 2A 0014E 26$:   SCANC  TEMP_LEN, @0(R8), TABLE, MASK         ; 2488
                                       02 12 00157        BNEQ   27$                                   ;
                                       51 D4 00159        CLRL   R1                                    ;
                           56          51 D0 0015B 27$:   MOVL   R1, SCAN_VAL                          ;
                                       03 12 0015E        BNEQ   28$                                   ; 2489
                                     0117 31 00160        BRW    45$                                   ;
                 16        06     08   AC CF 00163 28$:   CASEL  ELEM_TYPE, #6, #22                    ; 2491
   FFD3        0030      0030        0030    00168 29$:   .WORD  31$-29$,-                             ;
   FFD3        FFD3      0030        0030    00170               31$-29$,-                             ;
   FFD3        FFD3      FFD3        0081    00178               31$-29$,-                             ;
   0030        FFD3      FFD3        FFD3    00180               24$-29$,-                             ;
   FFD3        FFD3      FFD3        FFD3    00188               31$-29$,-                             ;
               0030      0030        FFD3    00190               31$-29$,-                             ;
                                                                 24$-29$,-                             ;
                                                                 24$-29$,-                             ;
                                                                 35$-29$,-                             ;
                                                                 24$-29$,-                             ;
                                                                 24$-29$,-                             ;
                                                                 24$-29$,-                             ;
                                                                 24$-29$,-                             ;
                                                                 24$-29$,-                             ;
                                                                 24$-29$,-                             ;
                                                                 31$-29$,-                             ;
                                                                 24$-29$,-                             ;
                                                                 24$-29$,-                             ;
                                                                 24$-29$,-                             ;
                                                                 24$-29$,-                             ;
                                                                 24$-29$,-                             ;
                                                                 31$-29$,-                             ;
                                                                 31$-29$                               ;
                                       A3 11 00196 30$:   BRB    24$                                   ;
                 59        56          68 C3 00198 31$:   SUBL3  (R8), SCAN_VAL, R9                    ; 2497
            04   BE        6A          59 28 0019C        MOVC3  R9, (PTRS), @PTRD                     ;
                           09          66 91 001A1        CMPB   (SCAN_VAL), #9                        ; 2498
                                       09 13 001A4        BEQL   32$                                   ;
                           20          66 91 001A6        CMPB   (SCAN_VAL), #32                       ; 2499
                                       04 13 001A9        BEQL   32$                                   ;
                                       66 95 001AB        TSTB   (SCAN_VAL)                            ; 2500
                                       1B 12 001AD        BNEQ   33$                                   ;
                      1C   AE          59 A0 001AF 32$:   ADDW2  R9, DSC                               ; 2510
                           5A   01   A94A 9E 001B3        MOVAB  1(R9)[PTRS], PTRS                     ; 2511
                      04   AE          59 C0 001B8        ADDL2  R9, PTRD                              ; 2512
                 50        57          59 C3 001BC        SUBL3  R9, LEN, R0                           ; 2513
                           57     FF   A0 9E 001C0        MOVAB  -1(R0), LEN                           ;
                           68     01   A6 9E 001C4        MOVAB  1(R6), (R8)                           ; 2514
                                       CC 11 001C8        BRB    30$                                   ; 2498
                           68          56 D1 001CA 33$:   CMPL   SCAN_VAL, (R8)                        ; 2518
                                       17 13 001CD        BEQL   34$                                   ;
```

```
                                                            M 14
BAS$$UDF_RL                                           16-Sep-1984 01:20:23    VAX-11 Bliss-32 V4.0-742                 Page  40
1-075                                                 14-Sep-1984 11:56:43    [BASRTL.SRC]BASUDFRL.B32;1                    (7)

                                     50      1C    AE  3C 001CF       MOVZWL  DSC, R0                                  ; 2529
                                     50            56  C0 001D3       ADDL2   SCAN_VAL, R0                             ;
              1C    AE               50            68  A3 001D6       SUBW3   (R8), R0, DSC                            ;
                    50               57            59  C3 001DB       SUBL3   R9, LEN, R0                              ; 2530
                                     57      FF    A0  9E 001DF       MOVAB   -1(R0), LEN                              ;
                                     68            56  D0 001E3       MOVL    SCAN_VAL, (R8)                           ; 2531
                                                 012D  31 001E6 34$:  BRW     55$                                      ; 2532
                                 08  AE            56  D0 001E9 35$:  MOVL    SCAN_VAL, A_HIGH_MARK                    ; 2552
                                     01      0C    AE  D1 001ED       CMPL    MASK, #1                                 ; 2553
                                                   15  12 001F1       BNEQ    38$                                      ;
                                     20      FF    A6  91 001F3 36$:  CMPB    -1(SCAN_VAL), #32                        ; 2555
                                                   0B  13 001F7       BEQL    37$                                      ;
                                     09      FF    A6  91 001F9       CMPB    -1(SCAN_VAL), #9                         ; 2556
                                                   05  13 001FD       BEQL    37$                                      ;
                                             FF    A6  95 001FF       TSTB    -1(SCAN_VAL)                             ; 2557
                                                   04  12 00202       BNEQ    38$                                      ;
                                                   56  D7 00204 37$:  DECL    SCAN_VAL                                 ; 2559
                                                   EB  11 00206       BRB     36$                                      ;
                    59               56            68  C3 00208 38$:  SUBL3   (R8), SCAN_VAL, R9                       ; 2561
                                 1C  AE            59  B0 0020C       MOVW    R9, DSC                                  ;
              04    BE               6A            59  28 00210       MOVC3   R9, (PTRS), @PTRD                        ; 2562
                                     68      08    AE  D0 00215       MOVL    A_HIGH_MARK, (R8)                        ; 2568
                                     27      08    BE  91 00219       CMPB    @A_HIGH_MARK, #39                        ; 2569
                                                   06  13 0021D       BEQL    39$                                      ;
                                     22      08    BE  91 0021F       CMPB    @A_HIGH_MARK, #34                        ;
                                                   52  12 00223       BNEQ    44$                                      ;
                                                   68  D6 00225 39$:  INCL    (R8)                                     ; 2576
                                 0C  AE      41    8F  9A 00227       MOVZBL  #65, MASK                                ; 2583
                                     50      1C    AE  3C 0022C       MOVZWL  DSC, R0                                  ; 2584
                    50               57            50  C3 00230       SUBL3   R0, LEN, R0                              ;
                                                   50  D7 00234       DECL    REM_LENGTH                               ;
                           00010000  8F            50  D1 00236       CMPL    REM_LENGTH, #65536                       ; 2585
                                                   05  1F 0023D       BLSSU   40$                                      ;
                                     50    FFFF    8F  3C 0023F       MOVZWL  #65535, REM_LENGTH                       ;
0C    AE      FCAD  CF               00  B8            50  2A 00244 40$:  SCANC   REM_LENGTH, @0(R8), TABLE, MASK          ; 2586
                                                   02  12 0024D       BNEQ    41$                                      ;
                                                   51  D4 0024F       CLRL    R1                                       ;
                                     50            51  D0 00251 41$:  MOVL    R1, T_RET_VAL                            ;
                                                   07  12 00254       BNEQ    42$                                      ; 2588
                    51           00  BE            01  C1 00256       ADDL3   #1, @0(SP), R1                           ;
                                                   03  11 0025B       BRB     43$                                      ;
                                     51            50  D0 0025D 42$:  MOVL    T_RET_VAL, R1                            ;
                                     68            51  D0 00260 43$:  MOVL    R1, (R8)                                 ;
                                                   50  D5 00263       TSTL    T_RET_VAL                                ; 2589
                                                   10  13 00265       BEQL    44$                                      ;
                                     2C            60  91 00267       CMPB    (T_RET_VAL), #44                         ; 2590
                                                   0B  13 0026A       BEQL    44$                                      ;
                                     7E     00G    8F  9A 0026C       MOVZBL  #BAS$K_DATFORERR, -(SP)                  ; 2591
                          00000000G  00            01  FB 00270       CALLS   #1, BAS$$STOP_IO                         ;
                                                 009C  31 00277 44$:  BRW     55$                                      ; 2593
                                     52      00    BE  D0 0027A 45$:  MOVL    @0(SP), T_BUF_END                        ; 2612
                                     08      0C    AE  D1 0027E       CMPL    MASK, #8                                 ; 2622
                                                   06  13 00282       BEQL    46$                                      ;
                                     04      0C    AE  D1 00284       CMPL    MASK, #4                                 ;
                                                   0B  12 00288       BNEQ    47$                                      ;
                                     7E     00G    8F  9A 0028A 46$:  MOVZBL  #BAS$K_DATFORERR, -(SP)                  ; 2624
                          00000000G  00            01  FB 0028E       CALLS   #1, BAS$$STOP_IO                         ;
```

```
                              59    FF71  CB 9A 00295 47$:   MOVZBL  -143(CCB), R9                       ; 2633
                              1E          59 91 0029A        CMPB    R9, #30                             ;
                                          05 13 0029D        BEQL    48$                                 ;
                              22          59 91 0029F        CMPB    R9, #34                             ; 2634
                                          1B 12 002A2        BNEQ    51$                                 ;
                              0E      08  AC D1 002A4 48$:   CMPL    ELEM_TYPE, #14                      ; 2635
                                          15 12 002A8        BNEQ    51$                                 ;
                              20      FF  A2 91 002AA 49$:   CMPB    -1(T_BUF_END), #32                  ; 2637
                                          0B 13 002AE        BEQL    50$                                 ;
                              09      FF  A2 91 002B0        CMPB    -1(T_BUF_END), #9                   ; 2638
                                          05 13 002B4        BEQL    50$                                 ;
                                      FF  A2 95 002B6        TSTB    -1(T_BUF_END)                       ; 2639
                                          04 12 002B9        BNEQ    51$                                 ;
                                          52 D7 002BB 50$:   DECL    T_BUF_END                           ; 2641
                                          EB 11 002BD        BRB     49$                                 ;
                              52          68 C2 002BF 51$:   SUBL2   (R8), R2                            ; 2643
                        1C    AE          52 A0 002C2        ADDW2   R2, DSC                             ;
         04      BE           6A          52 28 002C6        MOVC3   R2, (PTRS), @PTRD                   ; 2644
                        04    AE          53 D0 002CB        MOVL    R3, PTRD                            ;
                              20          59 91 002CF        CMPB    R9, #32                             ; 2645
                                          3E 12 002D2        BNEQ    54$                                 ;
                              50      0C  AB 3C 002D4        MOVZWL  12(CCB), R0                         ; 2663
                              1B          50 B1 002D8        CMPW    R0, #27                             ; 2665
                                          26 12 002DB        BNEQ    53$                                 ;
                              01      0E  AB B1 002DD        CMPW    14(CCB), #1                         ; 2672
                                          0B 12 002E1        BNEQ    52$                                 ;
                                      1C  AE B6 002E3        INCW    DSC                                 ; 2675
                        04    BE    FD0E  CF 90 002E6        MOVB    P.AAD, @PTRD                        ; 2676
                                          24 11 002EC        BRB     54$                                 ; 2672
                        1C    AE      0E  AB A0 002EE 52$:   ADDW2   14(CCB), DSC                        ; 2680
                              50      22  AB 3C 002F3        MOVZWL  34(CCB), R0                         ; 2681
                              50      28  AB C0 002F7        ADDL2   40(CCB), R0                         ;
         04      BE           60      0E  AB 28 002FB        MOVC3   14(CCB), (R0), @PTRD                ;
                                          0F 11 00301        BRB     54$                                 ; 2663
                              0D          50 B1 00303 53$:   CMPW    R0, #13                             ; 2684
                                          0A 12 00306        BNEQ    54$                                 ;
                        1C    AE          02 A0 00308        ADDW2   #2, DSC                             ; 2686
                        04    BE    FCEC  CF B0 0030C        MOVW    P.AAE, @PTRD                        ; 2687
                              68      00  BE D0 00312 54$:   MOVL    @0(SP), (R8)                        ; 2693
                        10    AE          01 D0 00316 55$:   MOVL    #1, RET_VAL                         ; 2694
                              36    FF71  CB 91 0031A        CMPB    -143(CCB), #54                      ; 2702
                                          07 13 0031F        BEQL    56$                                 ;
                              22    FF71  CB 91 00321        CMPB    -143(CCB), #34                      ; 2703
                                          0B 12 00326        BNEQ    57$                                 ;
                              50    FF48  CB D0 00328 56$:   MOVL    -184(CCB), BMF                      ; 2708
       0087      C0           68          01 C1 0032D        ADDL3   #1, (R8), 135(BMF)                  ; 2709
                 16           06      08  AC CF 00333 57$:   CASEL   ELEM_TYPE, #6, #22                  ; 2717
  012F         008E         0059        0031    00338 58$:   .WORD   60$-58$,-                           ;
  012F         012F         00C6        00A1    00340                61$-58$,-                           ;
  012F         012F         012F        0128    00348                64$-58$,-                           ;
  0128         012F         012F        012F    00350                75$-58$,-                           ;
  012F         012F         012F        012F    00358                66$-58$,-                           ;
               0110         00FB        012F    00360                67$-58$,-                           ;
                                                                     75$-58$,-                           ;
                                                                     75$-58$,-                           ;
                                                                     74$-58$,-                           ;
                                                                     75$-58$,-                           ;
```

```
                                                                75$-58$,-                               ;
                                                                75$-58$,-                               ;
                                                                75$-58$,-                               ;
                                                                75$-58$,-                               ;
                                                                75$-58$,-                               ;
                                                                74$-58$,-                               ;
                                                                75$-58$,-                               ;
                                                                75$-58$,-                               ;
                                                                75$-58$,-                               ;
                                                                75$-58$,-                               ;
                                                                75$-58$,-                               ;
                                                                70$-58$,-                               ;
                                                                72$-58$                                 ;
                                00FE   31 00366 59$:    BRW     75$                                     ; 2724
                                  05   DD 00369 60$:    PUSHL   #5                                      ; 2734
                                  04   DD 0036B         PUSHL   #4                                      ;
                             04   AC   DD 0036D         PUSHL   ELEM                                    ;
                             28   AE   9F 00370         PUSHAB  DSC                                     ;
            00000000G  00         04   FB 00373         CALLS   #4, OTS$CVT_TI_L                        ;
                       E9         50   E9 0037A         BLBC    R0, 59$                                 ;
            0000007F   8F    04   BC   D1 0037D         CMPL    @ELEM, #127                             ; 2741
                                  32   14 00385         BGTR    63$                                     ;
            FFFFFF80   8F    04   BC   D1 00387         CMPL    @ELEM, #-128                            ; 2742
                                  26   11 0038F         BRB     62$                                     ;
                                  05   DD 00391 61$:    PUSHL   #5                                      ; 2763
                                  04   DD 00393         PUSHL   #4                                      ;
                             04   AC   DD 00395         PUSHL   ELEM                                    ;
                             28   AE   9F 00398         PUSHAB  DSC                                     ;
            00000000G  00         04   FB 0039B         CALLS   #4, OTS$CVT_TI_L                        ;
                       C1         50   E9 003A2         BLBC    R0, 59$                                 ;
            00007FFF   8F    04   BC   D1 003A5         CMPL    @ELEM, #32767                           ; 2771
                                  0A   14 003AD         BGTR    63$                                     ;
            FFFF8000   8F    04   BC   D1 003AF         CMPL    @ELEM, #-32768                          ; 2772
                                  78   18 003B7 62$:    BGEQ    69$                                     ;
                       7E        00G   8F 9A 003B9 63$: MOVZBL  #BAS$K_ILLNUM, -(SP)                    ; 2774
            00000000G  00         01   FB 003BD         CALLS   #1, BAS$$STOP_IO                        ;
                                  11   11 003C4         BRB     65$                                     ;
                                  05   DD 003C6 64$:    PUSHL   #5                                      ; 2794
                                  04   DD 003C8         PUSHL   #4                                      ;
                             04   AC   DD 003CA         PUSHL   ELEM                                    ;
                             28   AE   9F 003CD         PUSHAB  DSC                                     ;
            00000000G  00         04   FB 003D0         CALLS   #4, OTS$CVT_TI_L                        ;
                                  6D   11 003D7 65$:    BRB     71$                                     ;
                       7E    7B   8F   9A 003D9 66$:    MOVZBL  #123, -(SP)                             ; 2803
                                  7E   7C 003DD         CLRQ    -(SP)                                   ;
                             20   AE   9F 003DF         PUSHAB  T_ELEM                                  ;
                             2C   AE   9F 003E2         PUSHAB  DSC                                     ;
            00000000G  00         05   FB 003E5         CALLS   #5, OTS$CVT_T_D                         ;
                       78         50   E9 003EC         BLBC    R0, 75$                                 ;
                             04   AC   DD 003EF         PUSHL   ELEM                                    ; 2804
                             18   AE   9F 003F2         PUSHAB  T_ELEM                                  ;
            00000000G  00         02   FB 003F5         CALLS   #2, LIB$CVTDF                           ;
                                  5D   11 003FC         BRB     73$                                     ;
                       7E    73   8F   9A 003FE 67$:    MOVZBL  #115, -(SP)                             ; 2815
                       7E  FF70   CB   98 00402         CVTBL   -144(CCB), -(SP)                        ;
                                  7E   D4 00407         CLRL    -(SP)                                   ;
                       52    04   AC   D0 00409         MOVL    ELEM, R2                                ;
```

```
                                        52  DD 0040D         PUSHL   R2                                    :
                              2C        AE  9F 0040F         PUSHAB  DSC                                   :
                00000000G  00           05  FB 00412         CALLS   #5, OTS$CVT_T_D                       :
                           53           50  D0 00419         MOVL    R0, STATUS                            :
                                FF70    CB  95 0041C         TSTB    -144(CCB)                             : 2819
                                        0C  13 00420         BEQL    68$                                   :
                                        52  DD 00422         PUSHL   R2                                    : 2822
                00000000G  00           01  FB 00424         CALLS   #1, MTH$DINT                          :
                           62           50  7D 0042B         MOVQ    R0, (R2)                              : 2827
                           36           53  E9 0042E 68$:    BLBC    STATUS, 75$                           : 2831
                                        3F  11 00431 69$:    BRB     76$                                   :
                           7E     73    8F  9A 00433 70$:    MOVZBL  #115, -(SP)                           : 2840
                                        7E  7C 00437         CLRQ    -(SP)                                 :
                                  04    AC  DD 00439         PUSHL   ELEM                                  :
                                  2C    AE  9F 0043C         PUSHAB  DSC                                   :
                00000000G  00           05  FB 0043F         CALLS   #5, OTS$CVT_T_G                       :
                                        13  11 00446 71$:    BRB     73$                                   : 2841
                           7E     73    8F  9A 00448 72$:    MOVZBL  #115, -(SP)                           : 2850
                                        7E  7C 0044C         CLRQ    -(SP)                                 :
                                  04    AC  DD 0044E         PUSHL   ELEM                                  :
                                  2C    AE  9F 00451         PUSHAB  DSC                                   :
                00000000G  00           05  FB 00454         CALLS   #5, OTS$CVT_T_H                       :
                           09           50  E9 0045B 73$:    BLBC    STATUS, 75$                           : 2851
                                        12  11 0045E         BRB     76$                                   :
                       04  BC     1C    AE  B0 00460 74$:    MOVW    DSC, @ELEM                            : 2860
                                        0B  11 00465         BRB     76$                                   :
                           7E    00G    8F  9A 00467 75$:    MOVZBL  #BAS$K_DATFORERR, -(SP)               : 2865
                00000000G  00           01  FB 0046B         CALLS   #1, BAS$$STOP_IO                      :
                           50     10    AE  D0 00472 76$:    MOVL    RET_VAL, R0                           : 2866
                                            04 00476         RET                                           :
                                        50  D4 00477 77$:    CLRL    R0                                    : 2867
                                            04 00479         RET                                           :

; Routine Size:  1146 bytes,    Routine Base:  _BAS$CODE + 0494


; 1578          2868  1     END
; 1579          2869  0 ELUDOM




;                               PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  _BAS$CODE                        2318  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)



;                              Library Statistics
;
;                                           -------- Symbols --------       Pages       Processing
;       File                                Total    Loaded  Percent        Mapped      Time
```

```
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1            9776        22          0        581            00:01.2



; Information:  2
; Warnings:     0
; Errors:       0


;                                         COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS$:BASUDFRL/OBJ=OBJ$:BASUDFRL MSRC$:BASUDFRL/UPDATE=(ENH$:BASUDFRL)

; 1580          2870  0
; Size:         2043 code + 275 data bytes
; Run Time:        00:45.4
; Elapsed Time:    01:40.2
; Lines/CPU Min:     3797
; Lexemes/CPU-Min: 25373
; Memory Used:  428 pages
; Compilation Complete
```

0032 AH-BT13A-SE DIGITAL EQUIPMENT CORPORATION
VAX/VMS V4.0 CONFIDENTIAL AND PROPRIETARY